
TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: Elektronické informační a řídicí systémy

Realizace ovládacího systému vrtulníku

Realization of Control System for Helicopter

Bakalářská práce

Autor: **Jiří Budasz**

Vedoucí práce: Ing. Petr Školník, Ph.D.

Konzultant: Ing. Lukáš Hubka, Ph.D.

V Liberci 18.5.2011





Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis



Abstrakt

Tato bakalářská práce se zabývá návrhem a konstrukcí ovládacího systému pro laboratorní model vrtulníku HUMUSOFT CE150. Nejdříve pojednává o stavu ovládacího systému a modelu na počátku práce. Poté je v práci diskutováno několik principů převodu signálů z inkrementálních čidel na napěťové signály. Dále je v práci rozebráno několik možností generování spínacího signálu pulzně šířkové modulace v závislosti na vstupním napětí. Práce se podrobně zabývá popisem návrhu a praktické konstrukce aplikovaného ovládacího systému. Zvolený systém se skládá ze dvou stejných na sobě nezávislých kanálů. Jedním je kanál azimut, se kterým je spojen postranní motor vrtulníku. Druhým je kanál elevace, ke kterému je připojen hlavní motor vrtulníku. Každý kanál převádí signál z inkrementálního čidla na napěťový signál informující o poloze. Na základě vstupního napětí kanál generuje signál v pulzně šířkové modulaci. Základem systému je použití jednočipových mikropočítačů s periferními obvody. K vyhodnocení signálů z inkrementálních čidel je použito čtyřnásobné interpolace, která vyhodnocuje každou hranu obou částí signálů čidel. Oba kanály mají svůj řídící mikropočítač, který zajišťuje tok všech informací, a to jak uvnitř tohoto mikropočítače, tak v periferních obvodech. Práce obsahuje podrobný popis obvodového zapojení i programů všech mikropočítačů, popis výroby plošného spoje s jeho osazením a zabudováním do přístroje. Práce též popisuje ověření funkčnosti zvoleného řešení měřením charakteristik ovládacího systému.

Klíčová slova: Model vrtulníku, HUMUSOFT CE150, pulzně šířková modulace, PWM, inkrementální čidla, mikropočítač.



Abstract

This bachelor thesis is about design and construction of a new control system for laboratory helicopter model HUMUSOFT CE150. The thesis describes current state of the model and its control system at the beginning. Several principles of conversion of signals from incremental encoder sensors to voltage signals are discussed in this thesis. Several possibilities of producing the pulse width modulation signal are mentioned. The thesis describes in details design and practical construction of the new control system. The system which has been chosen consists of two similar independent channels. The first channel is called Azimuth and is connected to the side motor of the helicopter. The main motor of the helicopter is connected to the second channel (Elevation). Every channel converts information from rotary encoder to output voltage which represents actual location of the model. The channel produces a signal for PWM on basis of input voltage. The main idea is using microcontrollers and their peripheral circuits. The circuit which deals with signals of rotary quadrature encoders uses a method which recognizes every slope of both parts of the signals. Both channels contain their own controlling microcontroller which deals with information inside microcontroller as well as in peripheral circuits. This thesis contains detailed description of the electrical circuit and description of programs running on all microcontrollers. The making of printed circuit board for the system is also described as well as its installation to the model. The functionality of the system was properly tested, behavioural characteristics were measured and the results are included at the end of this text.

Keywords: Model of helicopter, HUMUSOFT CE150, pulse width modulation, PWM, incremental encoders, microcontroller.



Obsah

Prohlášení.....	3
Abstrakt.....	4
Abstract	5
Obsah	6
Seznam obrázků	8
Seznam tabulek	8
1 Úvod	9
2 Stav modelu a ovládacího systému na začátku	11
3 Diskuze možných řešení ovládacího systému	14
3.1 Možnosti zpracování signálu z polohového čidla	14
3.2 Možnosti generování spínacího PWM signálu	17
4 Zvolené řešení ovládacího systému.....	20
4.1 Celkový popis zvoleného řešení	20
4.2 Popis obvodového zapojení	22
4.3 Vývoj programů pro mikropočítače	26
4.4 Program mikropočítače pro interpolaci.....	27
4.5 Program hlavního mikropočítače	28
5 Výroba a zabudování ovládacího systému	31
5.1 Návrh a výroba plošného spoje	31
5.2 Osazení a oživení desky plošných spojů	32
5.3 Zabudování ovládacího systému	34
6 Ověření funkčnosti ovládacího systému.....	36
6.1 Měření opakované přesnosti odměřování úhlů natočení.....	36
6.2 Měření převodních charakteristik odměřování úhlů natočení.....	37



6.3	Měření charakteristiky převodu napětí na signál PWM.....	39
7	Možnosti využití modelu a ovládacího systému	40
8	Závěr.....	41
	Použitá literatura	42
	Příloha A - Obsah přiloženého CD	43
	Příloha B - Schémata zapojení ovládacího systému.....	44
	Část zdrojů a konektorů	44
	Kanál azimut	45
	Kanál elevace	46
	Příloha C - Výpisy programů	47
	Program mikropočítače pro čtyřnásobnou interpolaci	47
	Program hlavního mikropočítače	49
	Příloha D - Deska plošných spojů	53
	Motiv spojů – spodní strana	53
	Motiv spojů – vrchní strana	54
	Rozmístění součástek – osazovací plán	55
	Seznam součástek	56
	Příloha E - Schéma připojení desky do zdroje	57
	Příloha F - Tabulky z měření	58
	Tabulky pro měření převodní charakteristiky odměřování.....	58
	Tabulka pro měření převodu napětí na PWM signál	59



Seznam obrázků

Obr. 1.1 Foto modelu vrtulníku s výkonovým zdrojem.....	9
Obr. 1.2 Označení pohybů v osách vrtulníku	10
Obr. 2.1 Záznam průběhu náběžné hrany signálu IRC čidla s rušením,.....	11
Obr. 2.2 Foto volného místa pro ovládací systém	13
Obr. 3.1 Princip čtyřnásobné interpolace.....	15
Obr. 3.2 Blokové schéma možného řešení generování PWM	18
Obr. 4.1 Blokové schéma 1 kanálu mého řešení.....	20
Obr. 4.2 Část zapojení se zdroji	22
Obr. 4.3 Část obvodu pro interpolaci.....	23
Obr. 4.4 Hlavní část zapojení kanálu s DA převodem.....	25
Obr. 4.5 Vývojový diagram programu hlavního mikropočítače kanálu	28
Obr. 5.1 Foto hotové desky plošných spojů.....	33
Obr. 5.2 Foto zabudovaného ovládacího systému	34
Obr. 6.1 Graf převodu odměřování v ose azimut s rovnicí získanou regresí.....	37
Obr. 6.2 Převod odměřování v ose elevace s rovnicí z regrese	38
Obr. 6.3 Graf převodu napětí na střidu PWM signálu – kanál azimut.....	39

Seznam tabulek

Tab. 4.1 Pravdivostní tabulka pro vyhodnocení směru.....	27
Tab. 5.1 Zapojení vývodů výstupního 37pinového konektoru	35
Tab. 6.1 Hodnoty z měření opakované přesnosti.....	36



1 Úvod

V této práci mám za úkol navrhnout a realizovat řešení ovládacího systému pro laboratorní model vrtulníku od firmy HUMUSOFT s.r.o. Tento model je určen k výuce v oborech zabývajících se automatickým řízením. Představuje nestabilní nelineární systém s několika vstupními akčními veličinami a několika výstupními veličinami. Celá sestava se skládá ze tří funkčních částí. Dvě z těchto částí jsou vidět na obr. 1.1, konkrétně to je samotný model vrtulníku se dvěma motory a vrtulemi. Tento



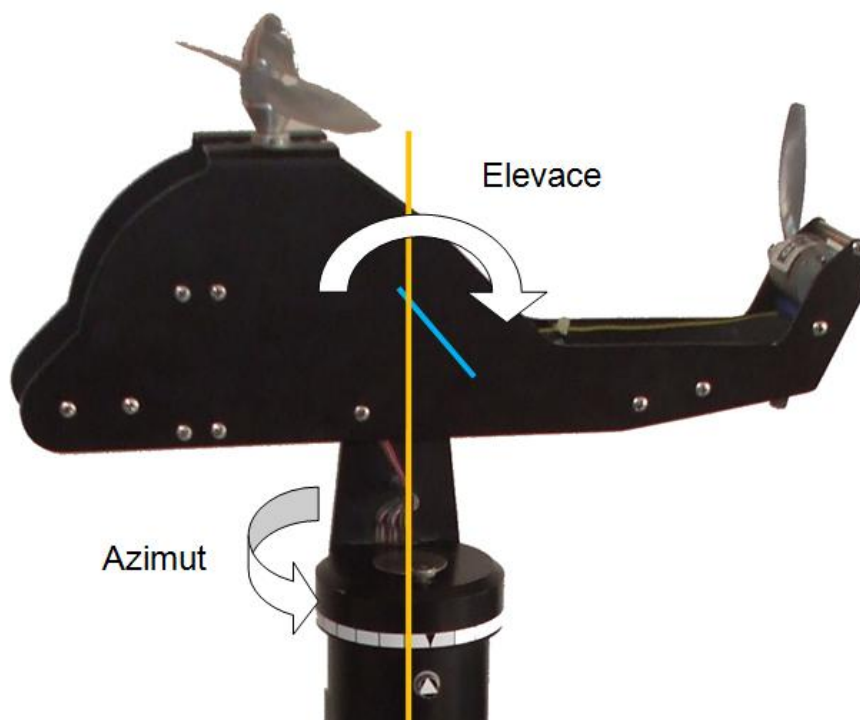
Obr. 1.1 Foto modelu vrtulníku s výkonovým zdrojem

model dále obsahuje dva inkrementální snímače natočení. Jeden snímá úhel natočení ve vertikální ose, tzv. azimut, a druhý snímá úhel v horizontální ose, neboli elevaci. Pohyby v těchto osách jsou vyznačeny na obr. 1.2, kde směr šipek vyznačuje kladný smysl pohybů. V zachycené situaci je model aretován v obou osách a oba úhly natočení jsou rovny nule. Model je pomocí otočného uložení připevněn k podstavci. Dalším dílem sestavy, který je vidět na obr. 1.1, je výkonový zdroj obsahující veškeré napájení



a výkonové budící obvody pro motory vrtulníku. Třetí částí modelu je řídicí karta realizující ovládací systém. Tato karta se umísťuje do počítače a umožňuje řídit vrtulník pomocí speciálního programu.

Zmíněná karta je zastaralá, neboť využívá ISA slotu, který není v dnešní době podporován. Firma HUMUSOFT s.r.o., dodavatel modelu, již nabízí nové řešení v podobě karty pro moderní PCI slot. Nicméně toto řešení je finančně dost náročné. Aby bylo možné model vůbec používat se současným vybavením, bylo třeba vytvořit pro model nový ovládací systém. S ním bude možné řídit vrtulník pomocí univerzální karty pro software MatLab, případně průmyslovým regulátorem. Toto je hlavní motivace mé práce.



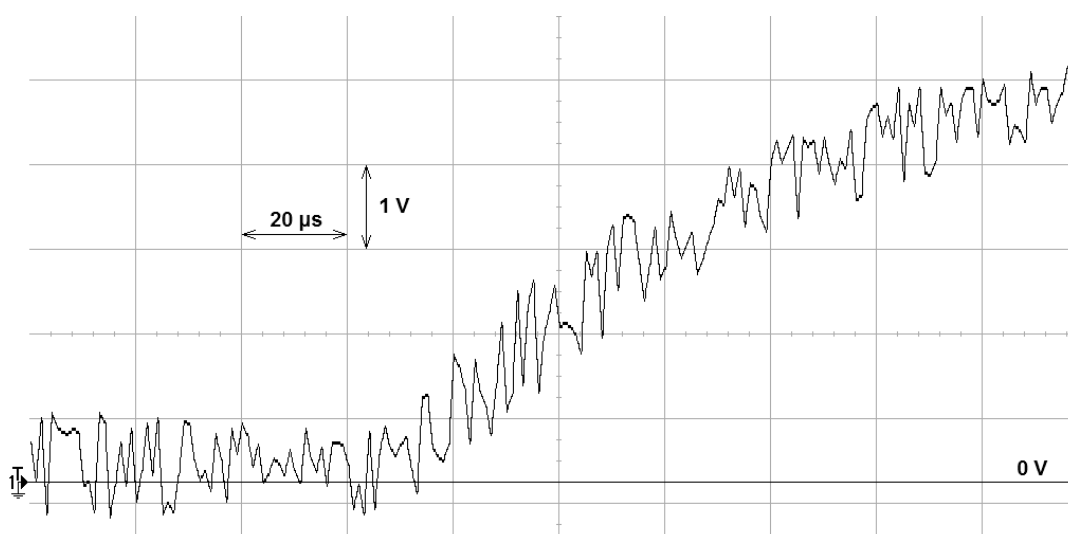
Obr. 1.2 Označení pohybů v osách vrtulníku



2 Stav modelu a ovládacího systému na začátku

Před započítím mých prací nebylo možné model používat. Výrobce dodávaný ovládací systém v podobě karty do počítače s rozhraním ISA nebylo možné se současnou technikou užívat. Existoval jistý pokus o realizaci ovládacího systému. Ten byl zkonstruován na velkém nepájivém poli a byl složen převážně ze základních logických bloků ve formě integrovaných obvodů. Tento pokus však také nebyl funkční. Patrně od jeho sestavení nebyl nijak udržován a chyběly některé vodiče. A protože neexistovala dokumentace k jeho zapojení, jakýkoliv pokus o jeho oživení jsem zavrhl. Vzhledem k poměrně značnému počtu integrovaných obvodů v zapojení a jeho velké nepřehlednosti jsem se ani nepokoušel o jakoukoliv jeho identifikaci.

Dále bylo nutné zhodnotit stav zbytku modelu, zjistit případné závady a hlavně získat veškeré parametry potřebné k návrhu a konstrukci ovládacího systému. Nejprve jsem prozkoumal technický manuál k modelu. Z [1] jsem zjistil zapojení všech konektorů a kabelů. Pak jsem ověřil pomocí měření průběhů výstupních signálů funkčnost inkrementálních snímačů snímajících úhel natočení v obou osách. Také jsem ověřil, že výstupní úrovně logických signálů ze snímačů odpovídají TTL logice. Toto bylo nutné, protože [1] jednoznačně nevypovídá o úrovních těchto výstupů. Ostatní digitální vstupy a výstupy by však měly mít úrovně kompatibilní s TTL logickými obvody. Změřil jsem též časový průběh signálu vystupujícího z inkrementálních čidel. Zjistil jsem ale závažnou komplikaci. Číslicový signál obsahoval velmi silné rušení, které je vidět na obr. 2.1. Závažnost problému se ukázala v měření následujícím. To spočívalo ve zjištění



**Obr. 2.1 Záznam průběhu náběžné hrany signálu IRC čidla s rušením,
měřeno osciloskopem Agilent MSO6012A**



počtu impulzů inkrementálních čidel na celý rozsah pohybu v jednotlivých osách. Čítač nebyl schopen spolehlivě toto změřit. Za výstupy čidel jsem musel umístit pasivní frekvenční filtry a Schmittovy klopné obvody. S touto modifikací bylo teprve možné spolehlivě změřit požadované hodnoty. Počet impulzů pro kanál azimut je 370 a pro elevaci je to 136. Při měření jsem objevil ještě jedno úskalí. To spočívá ve skutečnosti, že počet impulzů na rozsah pohybu se může lehce měnit v závislosti na rychlosti dojezdu ke krajní poloze. Tento jev je jistě způsoben pružnými dorazy v koncových polohách a setrvačností hmoty vrtulníku. Obě komplikace jsem musel při konstrukci systému zohlednit a eliminovat jejich vliv na funkci.

Dále bylo třeba zjistit, na kterém digitálním vstupu zdroje je povolovací signál pro běh motorů, tzv. enable bit. Tuto informaci jsem našel ručně dopsanou na jednom výtisku technického manuálu. Umístění zmíněného signálu v dokumentaci od výrobce není. Je to z jednoduchého důvodu, neboť výrobce předpokládal použití pouze s dodávaným ovládacím systémem. A tak s jeho propojením se zbytkem modelu uživatel nemusel být podrobně seznámen. Následně jsem zjišťoval možnosti napájení mého ovládacího systému ze zdroje pro model. K dispozici je tedy napájení napětím o velikosti 5 V, dále 12 V a symetricky i -12 V. Všechna napětí jsou proti potenciálu země modelu, která je galvanicky spojena s vodičem PE napájecí sítě. Dále jsem zjišťoval možnosti umístění ovládacího systému. Jednak bylo možné zkonstruovat ovládací systém jakožto další zařízení v samostatné krabici. To by se připojilo ke zdrojům a budičům přes 37pinový konektor. Ovládací systém by byl poté k nadřazenému řídicímu systému, v podobě např. počítačové karty pro MatLab, připojen pomocí dalšího konektoru vhodného typu.

Také jsem prozkoumal možnost zabudovat ovládací systém jako jednu desku plošných spojů do přístrojové krabice se zdrojem a budiči. Zjistil jsem, že v krabici je dostatečný prostor pro zabudování desky. Tu je možno připevnit pomocí distančních sloupků k jedné ze stěn krabice. Maximální rozměr desky je omezen na rozměr 120 mm × 190 mm. Jedinou komplikací při instalaci desky je fakt, že rovnoběžně k plánovanému umístění je plošný spoj obsahující poměrně velkou cívku. Bylo by tedy nutné v tomto případě desku doplnit o stínění. Ovládací systém by pak byl propojen uvnitř přístroje přímo se signálovými vodiči. S nadřazeným systémem by ovládání bylo propojeno



pomocí 37pinového konektoru, jehož zapojení by však bylo nutné pozměnit. Celá situace v přístrojové krabici, včetně volného místa, je vidět na obr. 2.2.

Po zhodnocení možností jsem se rozhodl směřovat konstrukci ovládacího systému dle druhé možnosti, a to zabudovat jej přímo do přístrojové krabice zdroje s budiči. Tato možnost bude výhodnější už jen proto, že vrtulník s podstavcem a krabice zdroje jsou samy o sobě rozměrné. Díky tomu je manipulace s nimi značně obtížná. Přidáním dalšího přístroje v krabici by se situace ještě zhoršila. Rozměry pro desku jsou dostatečné i pro relativně složitě zařízení na desce plošných spojů. V kritickém případě by se rozměry desky daly minimalizovat použitím technologií s malými rozměry součástek. Lze použít SMT (surface mount technology), což je technologie osazování součástek na plošný spoj, kdy součástky nemají vývody skrz desku, ale jsou připájeny k povrchu desky.



Obr. 2.2 Foto volného místa pro ovládací systém



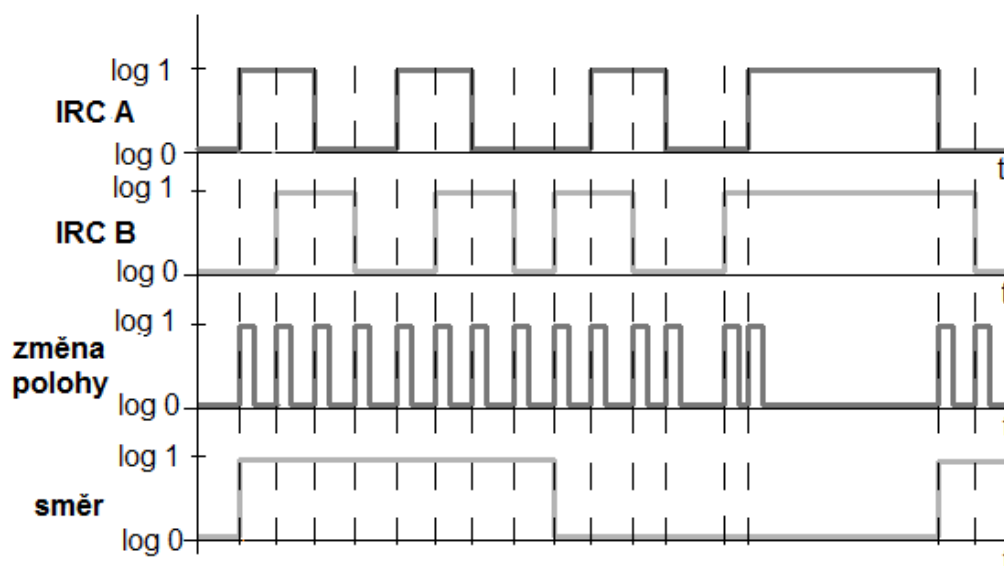
3 Diskuze možných řešení ovládacího systému

Zde bych rád uvedl základní myšlenku při návrhu ovládacího systému. Ta tkví v rozdělení celého ovládacího systému na dva totožné kanály. Jedním je kanál elevace a druhým azimut. Budu se tedy zabývat popisem pouze jednoho kanálu, přičemž oba kanály jsou stejné pouze s rozdílným převodem úhlu natočení na napěťový výstup. Stačí tedy navrhnout řešení jednoho kanálu a použít jej dvakrát. Dále tím bude zajištěno, že signály a informace příslušející jednotlivým kanálům se v průběhu zpracování nebudou nijak ovlivňovat. Díky tomu budou, z hlediska vyhodnocení a převodu, na sobě zcela nezávislé. Zde bych ještě zmínil, že hlavní motor vrtulníku bude ovládán kanálem nazvaným elevace a postranní motor bude ovládán kanálem azimut. Toto rozdělení jsem navrhl na základě fyzikální podstaty modelu, aby příslušný motor byl ovládán kanálem, jehož výchylku motor více ovlivňuje.

3.1 Možnosti zpracování signálu z polohového čidla

Samotný převod signálu lze rozdělit na dva podproblémy. Jedním je vyhodnocení signálů inkrementálního čidla, tzv. interpolace. Její výsledek bude nadále nutné převést z číslicového signálu na analogový. Budu tedy diskutovat možnosti samotné interpolace a poté i různé varianty převodu informace o poloze v číslicové podobě, ať již jako paralelní informace nebo jako sériové, na analogový signál.

Z inkrementálního rotačního čidla při pohybu měřené osy vycházejí dva signály, které jsou vzájemně posunuty o 90° . Pro vyhodnocení těchto signálů jsem vybral metodu tzv. čtyřnásobné interpolace, kdy se vyhodnocuje každá hrana z obou signálů čidla. Zvolená interpolace, jejíž princip je vidět na obr. 3.1, umožňuje získat přesnější informaci o poloze a její změně díky většímu rozlišení měření. Také umožňuje oproti jednonásobnému vyhodnocení, kdy je třeba celých impulsů každého signálu čidla, rychlejší informaci o změně polohy. Konstrukce jednonásobného interpolátoru by jistě byla velmi jednoduchá, nicméně vzhledem k možnostem dnešní elektroniky a také velké ztrátě informací při tomto způsobu vyhodnocení tuto metodu nebudu uvažovat.



Obr. 3.1 Princip čtyřnásobné interpolace

První a nejjednodušší možností je použít již hotový integrovaný obvod určený pro tuto funkci. Je jím obvod typu LS7084. Tento obvod dokonce podle [2] umožňuje přepínání mezi jednoduchou a čtyřnásobnou interpolací, avšak jak jsem již uvedl, použita by byla jen čtyřnásobná. Obvod LS7084 funguje dle obr. 3.1 pouze s tím rozdílem, že impulzní signál o přírůstku polohy je invertovaný. Tento obvod, byť by byl nejrychlejší cestou k cíli, však pro realizaci použít nelze, protože jej není možné sehnat. Patrně je již zastaralý, nebo po něm není poptávka, a tak jej nenabízí čeští ani zahraniční distributoři elektronických součástek.

Druhou možností realizace interpolace je sestavení logického obvodu skládajícího se ze základních bloků, jako klopné obvody a hradla. Signál o změně polohy v podobě krátkých impulzů by bylo možné generovat pomocí monostabilního klopného obvodu. Informaci o směru změny polohy by bylo možné získat pomocí sekvenčního obvodu, který by byl sestaven jako stavový automat typu Mealy, případně typu Moore s časováním pomocí náběžné hrany signálu o změně polohy. Další možnost volby spočívá v praktické konstrukci navrženého logického obvodu. Jednak by bylo možné sestavit obvod pomocí jednotlivých integrovaných obvodů. Tato možnost však má značnou nevýhodu v relativně velké složitosti výsledného zapojení a v porovnání s dalšími možnostmi také velkým počtem použitých integrovaných obvodů, a tudíž velkému zastavěnému prostoru na desce plošných spojů. Případně by bylo možné výslednou logickou funkci naprogramovat do obvodu FPGA (Field Programmable Gate



Array), což je logický obvod skládající se z programovatelných bloků propojených konfigurovatelnou maticí spojů. V tomto obvodu by bylo naprogramováno v oddělených funkčních blocích poté řešení celé číslicové části ovládacího systému. Posoudit výhody a nevýhody při případném použití FPGA obvodu tedy bude nutné v celkovém pohledu na návrh kompletního systému.

Třetí možností je řešení pomocí jednočipového mikropočítače. Celou logickou funkci by bylo možné sestavit jen naprogramováním sekvenčního programu. Zde se nabízejí dvě varianty. Jednou je, vložit interpolaci jen jako jednu z funkcí do mikropočítače, který by byl srdcem celého kanálu. Druhou variantou je použít samostatný mikropočítač, který by měl za úkol pouze provádět interpolaci a byl by tak v podstatě náhradou speciálního obvodu, o kterém byla zmínka výše. Dále by tento samostatný mikropočítač mohl vyhodnocovat výsledek interpolace a dokonce jej i předávat k dalšímu vyhodnocení, o kterém bude řeč později. V každém případě bude třeba řešit reakci programu na hranu signálu pomocí asynchronního přerušení a v obsluze tohoto přerušení pravděpodobně i vše vyhodnotit. Toto bude třeba provést co nejrychleji. Výhodou tohoto jsou mnohem menší nároky na počet součástek oproti realizaci stavového automatu integrovanými logickými obvody a také skutečnost, že funkci interpolace lze libovolně měnit jen změnou programu. Nevýhodou tohoto způsobu oproti předchozímu může být delší vyhodnocení jedné hrany a tím větší omezení maximální rychlosti reakce. To však bude možné částečně kompenzovat vysokým taktovacím kmitočtem mikropočítače.

Výsledek interpolace bude nutné převést na binární informaci o poloze. Řešení je velmi jednoduché. Spočívá v obousměrném čítání impulsů vycházejících z interpolace na základě informace o směru změny polohy. Řešení čítače v podstatě přímo navazuje na zvolený způsob interpolace. Pokud by byla řešena pomocí logických integrovaných obvodů, pak by i čítač byl takto řešen, nejlépe pomocí speciálního obvodu. Stejně tak pokud by byla logická funkce realizována v obvodu FPGA, byl by čítač její součástí. Rovněž v případě nasazení mikropočítače by byla funkce čítání integrována buď v interpolačním mikropočítači, nebo v jiném mikropočítači. Poslední otázkou, která byla v této části zatím nezodpovězena, je převod z binárního tvaru informace o poloze na analogový signál. Zde je v podstatě jediná racionální možnost, a to použití integrovaného digitálně analogového (dále jen D/A) převodníku. Volba spočívá v jeho



rozlišení. Pro toto zapojení a se současnou nabídkou na trhu bude 8 nebo 12 bitů. Dále lze volit zda způsob komunikace s D/A převodníkem bude paralelní nebo po sériovém rozhraní. Výhodou sériového přenosu je úspora vodičů a vývodů integrovaných obvodů. Nicméně tento způsob přenosu je možný pouze při použití mikropočítačů, které obsahují moduly pro komunikaci po sériových sběrnících. Pokud by se funkce realizovala pomocí logických bloků, tak by nejvýhodnější bylo použití paralelního přenosu mezi čítačem a D/A převodníkem. Použití sériového přenosu by zapojení komplikovalo, obzvláště při použití integrovaných logických obvodů. Výhodou paralelního přenosu může být rychlost. Nicméně bude-li délka sériové sběrnice krátká a díky tomu i frekvence jejího hodinového signálu bude moci být vysoká, v řádu MHz, rozdíl v rychlostech nebude v této aplikaci znatelný.

3.2 Možnosti generování spínacího PWM signálu

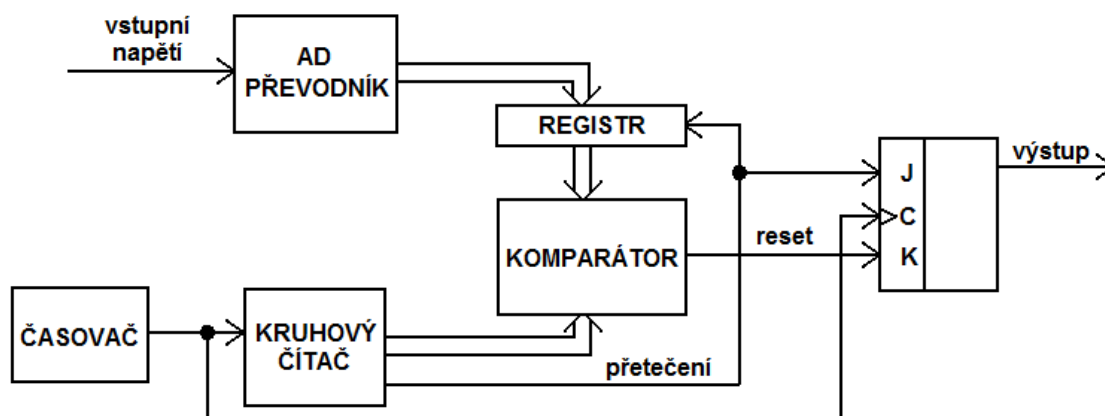
Kanál ovládacího systému musí obsahovat převod vstupního napětíového signálu 0 – 10 V na výstupní spínací signál pulzně šířkové modulace pro výkonové budiče motorů. Parametry výstupního signálu musí splňovat úroveň pro TTL logiku. Frekvence výstupního signálu musí být bezpodmínečně vyšší než rozsah kmitočtů slyšitelných lidským uchem, aby nedocházelo při provozu zařízení k nežádoucímu a obtěžujícímu pískání motorů. Vzhledem k tomu, že tento převod je jednodušší, než předchozí princip, je i tato část stručnější než předcházející.

První možností je analogová konstrukce pomocí operačních zesilovačů. Zapojení by obsahovalo generátor pilovitého průběhu. Generátor by se skládal z astabilního klopného obvodu a dále z integrátoru, na jehož výstupu by byl právě požadovaný pilovitý signál. Takto získaný průběh by se v komparátoru porovnával se vstupním napětím. Podle velikosti vstupního napětí by komparátor spínal v okamžicích, kdy bude toto napětí shodné s okamžitým napětím pilového průběhu. A naopak vypínal při stejné situaci ale při opačném smyslu změny napětí pily. Z takto fungujícího komparátoru by byl výstupem požadovaný signál PWM závislý na vstupním napětí. U zmíněného komparátoru by bylo jistě třeba dodat nějakou, byť malou, hysterezi, aby ve výstupním signálu nebyly nežádoucí zákmity. Dále by bylo třeba dbát na požadované výstupní napětí, což lze řešit například nesymetrickým napájením odpovídající velikosti. Tento způsob řešení však v sobě skrývá několik problémů. Jednak by zapojení obsahovalo



minimálně tři operační zesilovače, což znamená alespoň dva integrované obvody. Znamenalo by to také velké množství diskrétních součástek, které by byly kolem těchto operačních zesilovačů zapojeny. Komplikace s rozměry lze elegantně řešit použitím součástek pro povrchovou montáž SMD (surface mount device) a tím celé zapojení významně minimalizovat. Nicméně ani použití SMD součástek nevyřeší jiný problém. Ten tkví ve skutečnosti, že funkce a parametry výstupního signálu jsou závislé na hodnotách právě těchto součástek. Například frekvence a velikost použitého pilového signálu je závislá na konstantě integrátoru a časovacím obvodu v astabilním klopném obvodu. To znamená jednak, že veškeré nastavení takto konstruovaného obvodu lze provádět volbou nebo změnou hodnot konkrétních součástek. Dále to znamená, že celý obvod bude jen tak stabilní, jak stabilní budou hodnoty těchto součástek.

Další možností generování PWM signálu, jak je blokově zobrazeno na obr. 3.2, je nejprve analogový signál převést pomocí analogově digitálního (dále jen A/D) převodníku. Tato digitální informace by poté byla vstupem logického obvodu. Ten by obsahoval jednak generátor hodinových impulzů. Dále by obsahoval jednoduchý čítač čítající neustále dokola dle taktovacích impulzů z generátoru. Počet bitů čítače by musel odpovídat počtu bitů výstupního slova z A/D převodníku. Obvod by dále musel



Obr. 3.2 Blokové schéma možného řešení generování PWM

obsahovat komparátor a klopný obvod. Na začátku každého cyklu čítače, tedy při nulovém stavu, by se výstupní paměťový obvod, například hranový klopný obvod typu D nebo klopný obvod typu JK, nastavil do logické 1. Komparátor by porovnával aktuální stav čítače a binární hodnotu vstupního napětí. V okamžiku, kdy hodnoty budou stejné, by komparátor resetoval výstupní obvod. Toto by se dělo v každém cyklu čítače.



Tím je také určeno, že perioda cyklů čítače je přímo periodou výstupního signálu. Obvod by patrně musel být ještě rozšířen o záchytný registr pro binární hodnotu, pokud by ho neobsahoval A/D převodník, aby nedocházelo ke změně hodnoty binární informace během jednoho cyklu čítače. Tato hodnota by měla být vždy aktualizována na začátku cyklu čítače.

Tento způsob realizace by oproti předchozímu, díky použití číslicových obvodů, vyřešil do velké míry časovou a teplotní nestálost. Prakticky by se tento obvod dal opět řešit pomocí jednotlivých integrovaných obvodů. Znamenalo by to ale velké množství pouzder a s tím související velký zabraný prostor. Lze jej též řešit pomocí jednoho A/D převodníku v podobě integrovaného obvodu a celá číslicová část by mohla být naprogramována opět v obvodu typu FPGA. Jeho nasazení by však, jak jsem již dříve uvedl, muselo být mnohem komplexnější. Případně by bylo také možné navázat na A/D převodník specializovaným obvodem pro generování PWM signálu. Podle [3] by to mohl například být obvod typu LTC6992. Tento typ umožňuje velikostí vstupního napětí plynule měnit střidu výstupního PWM signálu. Zmíněný obvod by mohl přímo navazovat na A/D převodník pouze s děličem omezujícím napětí. Dále by podle [4] bylo možné použít obvod typu CDP68HC68W. Nicméně u tohoto obvodu se provádí zápis dat pomocí sériové sběrnice SPI (Serial Peripheral Interface), což je speciální sběrnice určená především pro komunikaci mikropočítačů s periferiemi. Z toho vyplývá, že by bylo nutné zajistit přenos dat z A/D převodníku po této sběrnici.

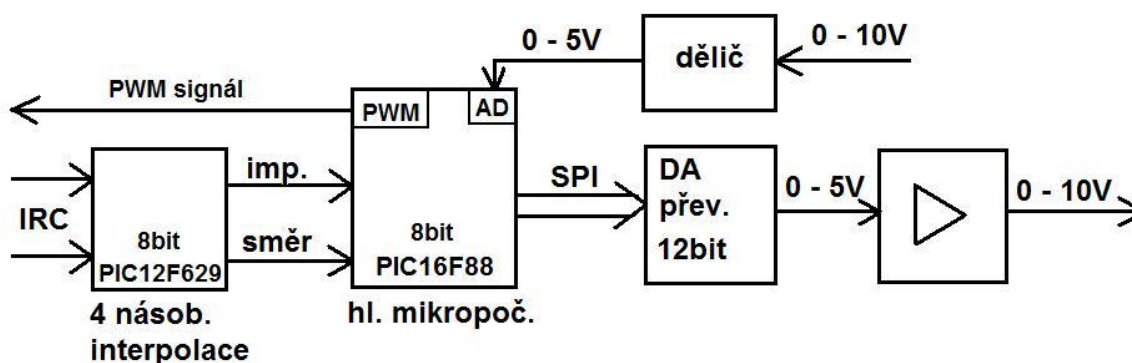
Jinou možností je využití mikropočítače, který by obsahoval A/D převodník. Na základě výstupu z tohoto převodníku by generoval signál pro PWM. Tato funkce by byla snadno realizovatelná s mikropočítačem, který obsahuje speciální PWM modul. V takové situaci se konstrukce celého převodu omezuje v podstatě jen na několik řádků programu mikropočítače. Toto řešení je oproti předchozím velmi jednoduché a přímočaré. Má minimální nároky na počet použitých součástek, díky tomu i na rozměry, a jediná nevýhoda by mohla být způsobena nekonzistentními vlastnostmi A/D převodníku vzhledem k teplotě a času. Touto je však zatížena zhruba ve stejné míře předchozí varianta, o první analogové ani nemluvě. Toto řešení je tak jednoduché, že by bylo jistě zbytečné použít jeden mikropočítač pouze pro realizaci jedné funkce. Výhodnější se jeví spíše varianta naprogramovat tuto funkci do mikropočítače, který bude řešit více úkolů v rámci jednoho kanálu a bude například tvořit jakési srdce kanálu.



4 Zvolené řešení ovládacího systému

4.1 Celkový popis zvoleného řešení

Na základě diskuze obsažené v předchozí kapitole jsem zvolil následující řešení. Blokové schéma jednoho kanálu je vidět na obr. 4.1. Srdcem kanálu je 8bitový



Obr. 4.1 Blokové schéma 1 kanálu mého řešení

jednočipový mikropočítač typu PIC16F88. Ten obstarává funkci převodu z napětíového řídicího vstupu na výstup pro PWM signál. Vzhledem k tomu, že použitý typ mikropočítače obsahuje integrované moduly A/D převodníku i generátoru PWM signálu, je implementace této části velmi snadná.

Dále je úkolem hlavního mikropočítače čítání pulzů vypovídajících o změně polohy, jejich zpracování, přepočet na správný rozsah a následné odeslání do D/A převodníku po sériové sběrnici SPI. Tento typ mikropočítače obsahuje integrovaný modul pro obsluhu této sběrnice. Zmíněný D/A převodník je typu MCP4921. Tento je 12bitový, a tudíž má výstup velmi dobré rozlišení. Pro rozsah 0 – 10 V je to 2,44 mV, počítám-li s navazujícím výstupním zesilovačem popsáným dále. Použitý převodník má však dle [5] nevýhodu, která tkví v maximálním dovoleném napájecím napětí 5,5 V. Díky tomu je to i maximum výstupního napětí. Proto je za něj ještě umístěn zesilovač se zesílením o velikosti 2 převádějící signál z rozsahu 0 – 5 V na rozsah 0 – 10 V. Výhodou tohoto typu převodníku je naopak skutečnost, že dokáže pracovat i při nesymetrickém napájení.

Poslední dosud nezmíněnou funkcí je interpolace signálu z IRC čidel. Řešena je odděleně od hlavního mikropočítače, aby byla zajištěna spolehlivá a rychlá reakce na každou hranu. Zvolený typ hlavního mikropočítače má také omezenou možnost



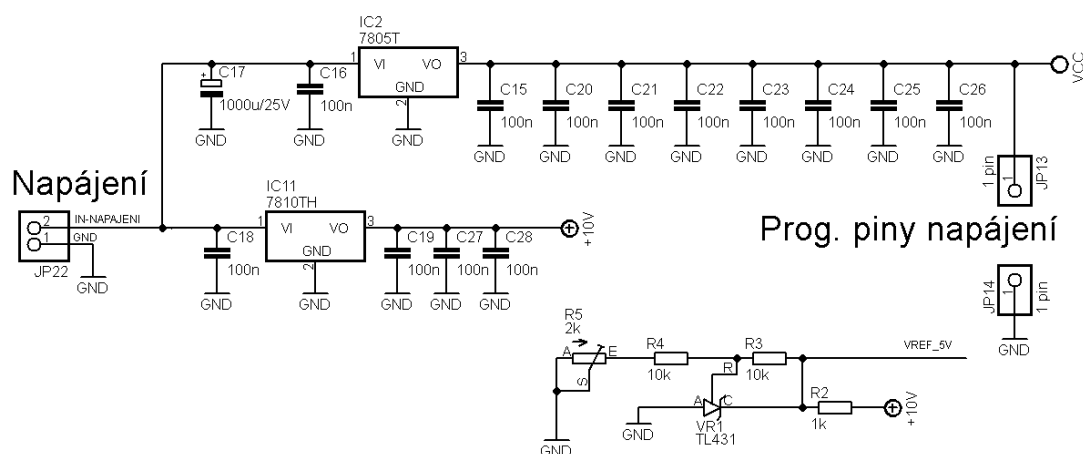
generování asynchronního přerušení od více vstupů zároveň. U tohoto typu nelze, s jednou výjimkou vývodu RB0, povolit nebo zakázat přerušení pouze od jednotlivých pinů. Funkci interpolace tedy zajišťuje 8bitový jednočipový mikropočítač typu PIC12F629. Ten zajišťuje čtyřnásobnou interpolaci popsanou v kapitole 3.1. Volba padla na uvedený typ díky menšímu počtu vývodů. Zároveň umožňuje generování asynchronního přerušení programu na základě jakékoliv hrany logického signálu na vstupech. V neposlední řadě také pro zvolený typ hovoří i jeho příznivá cena.

Krátce bych se ještě rád zmínil o důvodech pro výběr popsané možnosti realizace. Jednak mám s návrhem a konstrukcí obvodů tohoto typu zkušenosti. Také vlastním fyzické prostředky pro vývoj podobných aplikací, hlavně pro programování mikropočítačů. Další výhodou tohoto zapojení je nepochybně skutečnost, že větší část funkčnosti je dána programem jednotlivých mikropočítačů. To šetří jak náklady, tak i zastavěný prostor výsledného zařízení a díky tomu lze též později funkci modifikovat dle aktuální potřeby bez zásahu do zapojení. Srovnám-li mnou vybrané řešení s ostatními možnostmi, kromě použití FPGA obvodu, pak ve prospěch mého návrhu hovoří menší počet potřebných pouzder. Také je potřeba menší počet diskrétních součástek přímo ovlivňujících funkci případně parametry zařízení.

Oproti tomu možnost realizace pomocí obvodu FPGA, by velmi redukovala počet použitých pouzder. Jistě by zde byla i výhoda daná tím, že většina funkce by byla naprogramována. Nevýhodou je však vyšší cena než u mikropočítačů. Dále je zde několik konstrukčních komplikací. První vyplývá z použití FPGA s volatilní konfigurací, kde by bylo třeba zajistit pomocí dalších obvodů naprogramování konfigurace do obvodu po každém zapnutí napájecího napětí. Druhou komplikací, která je v podstatě s popisovanou aplikací prakticky neslučitelná, je typ pouzdra FPGA obvodů. Tato pouzdra mají mnoho vývodů a jsou vyráběna pro technologii povrchového osazování, tzv. SMT. Vývody jsou často v podobě plošek zespoda pouzdra. Ta se vyznačují velmi malými rozměry, a tak je nemožné tyto součástky osazovat a pájet k plošnému spoji ručně. Bylo by tedy třeba použít pokročilou technologii osazování, v tomto případě nejspíše přetavením v horkovzdušné peci. To vše je finančně značně náročné a též by vznikly velké nároky na návrh a výrobu plošného spoje. Vzhledem k tomu, že výsledkem mé práce má být jedno fungující zařízení bez jakékoliv naděje na sériovou výrobu

4.2 Popis obvodového zapojení

První částí obvodu je zdroj. Na obr. 4.2 je vidět zdrojová část zapojení. Vstupní napájecí napětí o velikosti 12 V je nejdříve přivedeno na filtrační kondenzátor C17, který má za úkol odstranit zvlnění napětí. Napětí je dále vedeno na dva lineární třísvorkové

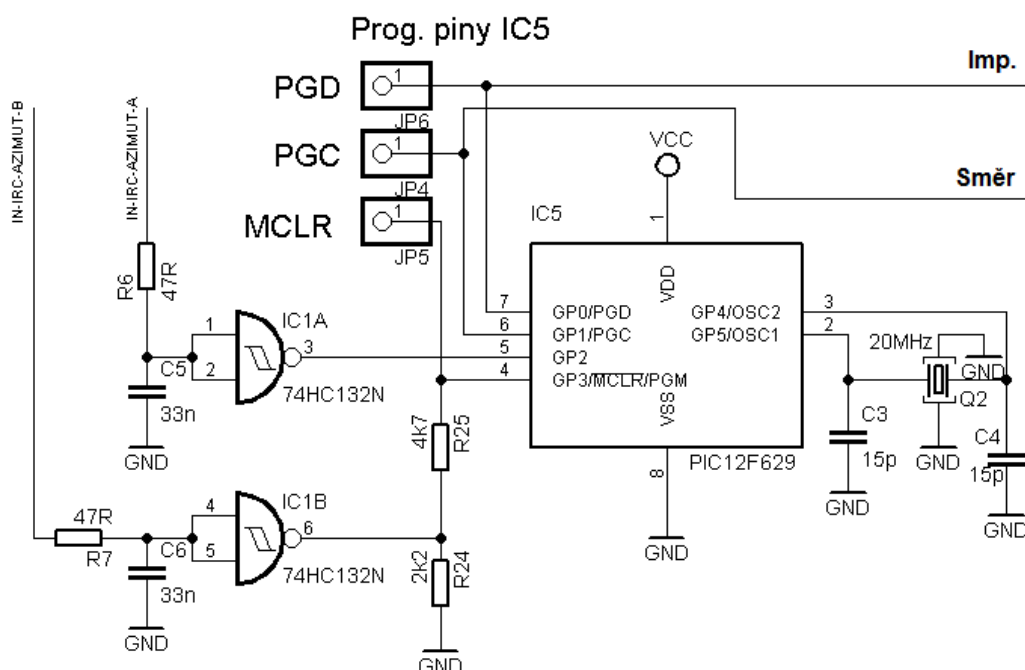


Obr. 4.2 Část zapojení se zdroji

stabilizátory, IC2 a IC11. Zmíněné obvody jsou v podstatě ve svém katalogovém zapojení. Z jejich výstupů se odebírají napětí 5 V a 10 V. Kondenzátory C20 až C28 jsou blokovací. Jsou nakonec umístěny přímo u napájení jednotlivých integrovaných obvodů. Zapojení dále obsahuje dva programovací piny, jejichž význam bude spolu s ostatními popsán později. Nakonec je na obr. 4.2 zobrazen zdroj přesného referenčního napětí o velikosti 5 V pro D/A převodník. Jedná se opět o katalogové zapojení integrovaného stabilizátoru TL431, který je často označován jako nastavitelná Zenerova dioda. Zde bych pouze uvedl, že trimrem R5 lze požadované napětí přesně nastavit.



Další částí zapojení je interpolátor signálu z IRC čidel. Jeho zapojení vidíme na obr. 4.3. Oba signály z IRC čidla jsou nejprve filtrovány pasivními filtry, skládajícími se z R6 a C5, potažmo R7 a C6. Vzhledem k velké vstupní impedanci Schmittova



Obr. 4.3 Část obvodu pro interpolaci

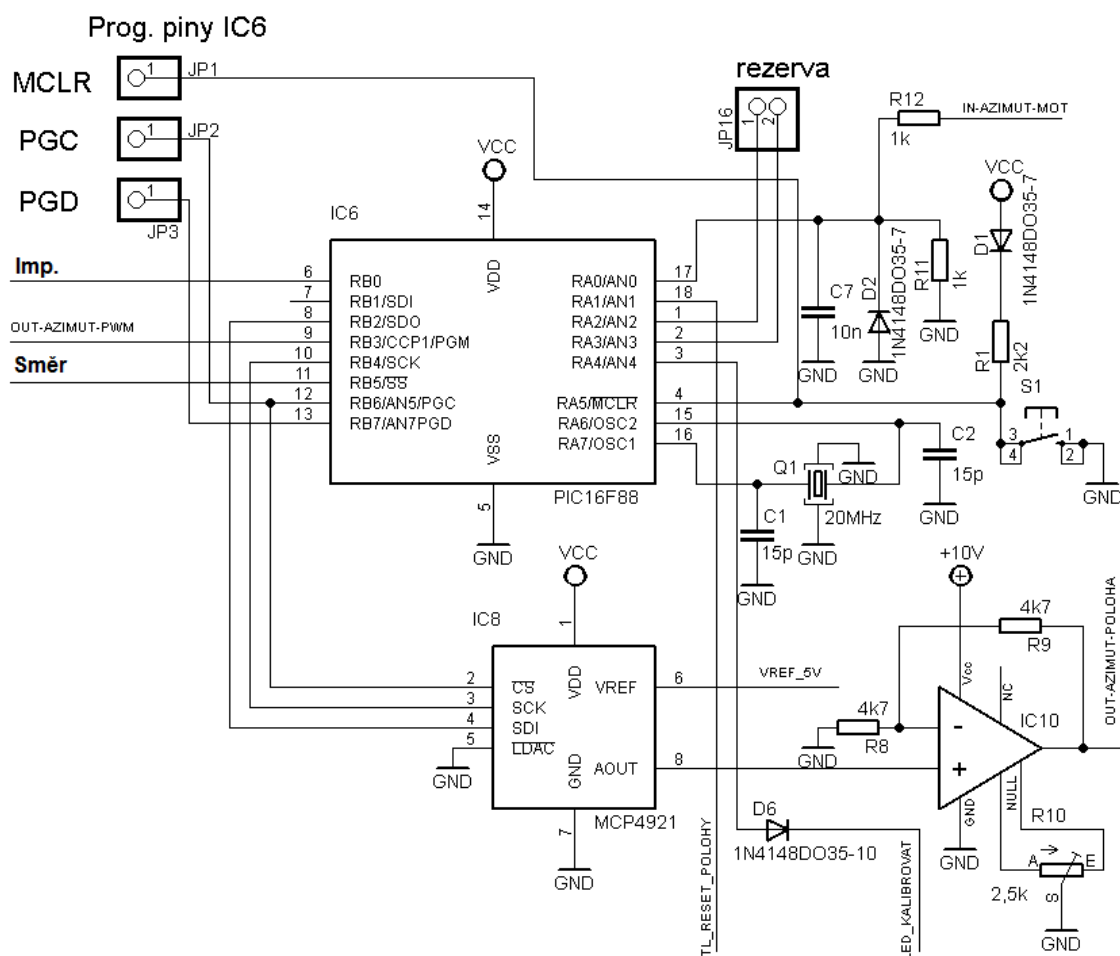
klopného obvodu následujícího za filtry a nízké hodnotě odporů rezistorů, filtr nepředstavuje pro užitečné signály žádnou překážku. Naopak rušení o vyšší frekvenci, o kterém byla zmínka v kapitole 2, účinně eliminuje. Zlomová frekvence filtrů je 103 kHz. Z výstupů Schmittova klopného obvodu jdou upravené signály do mikropočítače IC5, který zajišťuje 4násobnou interpolaci. Výsledek je poté veden do další části obvodu. Krátce bych se ještě zmínil o pomocných obvodech kolem mikropočítače. Jednak je zde časovací obvod složený z krystalu Q2 a kondenzátorů C3 a C4 a dále piny pro zavádění programu do mikropočítače. K tomu jsou dále potřeba piny pro vyvedení napájení, o kterých již byla zmínka. Navržené zapojení umožňuje použití technologie ICSP (In Circuit Serial Programming), kterou firma Microchip implementuje do svých mikropočítačů a která je určena k programování těchto obvodů přímo v hotové aplikaci bez nutnosti vyjímání ze zapojení. Zatím jsem úmyslně přecházel význam rezistorů R24 a R25. Ten souvisí právě se zaváděním programu, při kterém je na vstup MCLR (Masterclear) mikropočítače přivedeno napětí o velikosti 13 V. To by mohlo zničit výstup hradla IC1B. Aby se tak nedělo, jsou zde právě tyto rezistory fungující



při programování jako dělič napětí. Při normální funkci obvodu rezistory nepředstavují žádnou překážku vzhledem k dostatečné zatížitelnosti výstupu hradla a velké vstupní impedanci vstupu mikropočítače.

Nyní bych se věnoval zbytku zapojení zachycenému na obr. 4.4. Obsahuje především srdce celého kanálu v podobě mikropočítače IC6. Ten zpracovává výstupy z 4násobné interpolace, na jejichž základě posílá informace do D/A převodníku IC8. Komunikace, jak již bylo výše zmíněno, probíhá po sběrnici SPI, která se v tomto případě skládá ze tří vodičů. Je to datový vodič SDO/SDI, časovací SCK a nakonec výběrový CS. Obvod IC8 na základě vstupních dat a referenčního napětí o velikosti 5 V generuje odpovídající výstupní napětí. Následně je zesíleno výstupním neinvertujícím zesilovačem se zesílením o velikosti 2. Ten je tvořen především operačním zesilovačem IC10, typu AD820, který je napájen nesymetrickým napětím 10 V. Způsob napájení je zvolen z důvodu, aby se nikdy nemohlo na výstupu zařízení objevit napětí mimo rozsah 0 - 10 V, a to ani při zničení IC10. Podle [6] nejen umí zvolený typ pracovat i při nesymetrickém napájení, ale také má výstup tzv. rail to rail, což znamená, že výstupní napětí téměř dosahuje napájecího. U použitého typu je v daném zapojení podle [6] maximální rozdíl mezi výstupním saturačním napětím a napájením maximálně 40mV při nezatíženém výstupu nebo s velkou zatěžovací impedancí. To je v podstatě cena za bezpečnost vstupů navazujícího zařízení. Uvedená daň bezpečnosti je však jistě přijatelná obzvláště vzhledem k cenám používaných zařízení. Vedle IC10 ještě nalézáme trimr R10, kterým lze kompenzovat vstupní napětíovou nesymetrii operačního zesilovače.

Vrátím-li se k popisu funkce hlavního mikropočítače IC6 na obr. 4.4, je vidět, že také zpracovává vstupní napěťový signál. Ten je však nejprve pomocí jednoduchého děliče vydělen 2 do potřebného rozsahu. Dioda D2 chrání vstup mikropočítače před zničením, kdyby došlo k přepólování vstupního napětí. Kondenzátor C7 spolu s děličem tvořeným R11 a R12 funguje jako jednoduchý filtr vyšších frekvencí. Na základě takto upraveného napětí mikropočítač generuje výstupní PWM signál, který



Obr. 4.4 Hlavní část zapojení kanálu s DA převodem

vede přímo na výstup obvodu. Do IC6 dále vstupuje signál z tlačítka pro kalibraci odměřování úhlu. Naopak z něj vystupuje signál pro signalizační LED diodu informující právě o nutnosti kalibrace odměřování. Dále zapojení obsahuje konektor JP16, který zatím neplní žádnou funkci. Umožňuje v budoucnu dle potřeby připojit další zařízení. Piny, na které je konektor připojen, mohou být nastaveny jako digitální vstup nebo výstup, případně jako analogový vstup. To umožňuje velkou variabilitu rozšíření.



Nakonec tato část obvodu opět obsahuje pomocné obvody mikropočítače. Jednak je to časovací krystal Q1 s kondenzátory, dále jsou to programovací piny se stejnou funkcí jako u předchozího popisovaného mikropočítače. Pak je tu ještě resetovací obvod, jehož hlavní částí je tlačítko S1. Jeho stisknutím se resetuje běh programu mikropočítače. Rezistor R1 zajišťuje přítomnost log. 1 na resetovacím vstupu MCLR. Dioda D1 zabezpečuje, aby při zavádění programu se napětí o velikosti 13 V nespojilo ani přes R1 na napájení 5 V.

4.3 Vývoj programů pro mikropočítače

Aby zapojení popisované v kapitole 4.2 fungovalo dle potřeby, bylo nutné naprogramovat všechny jednočipové mikropočítače. Pro potřeby programování a odzkoušení celkové funkce jsem si na nepájivém poli složil zapojení pro jeden kanál. Pouze nebyl zapojen výstupní zesilovač. V tomto obvodu jsem poté ladil programy i detaily zapojení. Jak jsem již výše uváděl, celé řešení se skládá ze dvou stejných kanálů, což se samozřejmě promítlo i do tvorby softwaru. Bylo tedy třeba vytvořit dva programy pro mikropočítače, přičemž kód nahráný do hlavních mikropočítačů každého kanálu, tedy IC6 a IC7, je totožný až na jednu číselnou konstantu udávající převod počtu impulzů na výstupní napětí. Zmíněný kód jsem vyvíjel v jazyce C. Použití assembleru jsem vzhledem k možnostem překladačů a knihoven pro jazyk C v tomto případě ani neuvažoval. Použití jazyka C umožňuje mnohem rychlejší ale hlavně přehlednější a robustnější tvorbu programu, o práci s datovými formáty pro desetinná čísla ani nemluvě. K samotné tvorbě a softwarovému ladění kódu jsem použil program MPLAB, který je volně dostupný v poměrně slušně vybavené verzi. Pro překlad jsem použil kompilátor HI-TECH PICC, který je pro nekomerční užití opět zdarma dostupný. Vzhledem k tomu, že příslušné kódy programu jsou dosti dlouhé, nebudu je v následujících kapitolách uvádět. Omezím se pouze na jejich popis, při kterém se spíše zaměřím na celkovou funkci a chování mikropočítače. Nebudu se zabývat popisem každého řádku programu. Pro takový popis je v příloze C výpis obou programů s podrobným komentářem.



4.4 Program mikropočítače pro interpolaci

Nejdříve jsem vytvářel program pro mikropočítač PIC12F629, který má za úkol provádět čtyřnásobnou interpolaci. Struktura výsledného kódu je velmi jednoduchá. Po zavolání funkce pro počáteční nastavení všech parametrů mikropočítače je program v nekonečné smyčce, kde pouze v každém cyklu nuluje čítač watchdog, který je ochranou proti zacyklení. Mikropočítač tak tedy pouze čeká na asynchronní přerušení od hrany logického signálu na vstupech obvodu. Celá funkce mikropočítače je prakticky obsažena v kódu funkce pro obsluhu právě tohoto přerušení. Na začátku zmíněné funkce si mikropočítač uloží současný stav vstupů tak, že aktuální hodnoty budou obsaženy ve zvláštní proměnné jako 2. a 3. bit, přičemž přebírám obvyklou mnemotechniku a indexy bitů začínám indexem 0. V této proměnné jsou obsaženy v bitech 0 a 1 předchozí hodnoty vstupů. Hodnota této proměnné je vyhodnocena příkazem switch dle tab. 4.1, kde jsou uvedeny všechny možné vzniklé stavy před a po hraně signálu. Vzhledem k tomu, že vstupům předřazená hradla jsou invertující, jsou stavy zapsány

Tab. 4.1 Pravdivostní tabulka pro vyhodnocení směru

Stav nyní		Předchozí stav		stav hexadecimálně	směr
B	A	B	A		
1	1	1	0	0x0E	1
1	0	0	0	0x08	1
0	0	0	1	0x01	1
0	1	1	1	0x07	1
1	1	0	1	0x0D	0
0	1	0	0	0x04	0
0	0	1	0	0x02	0
1	0	1	1	0x0B	0

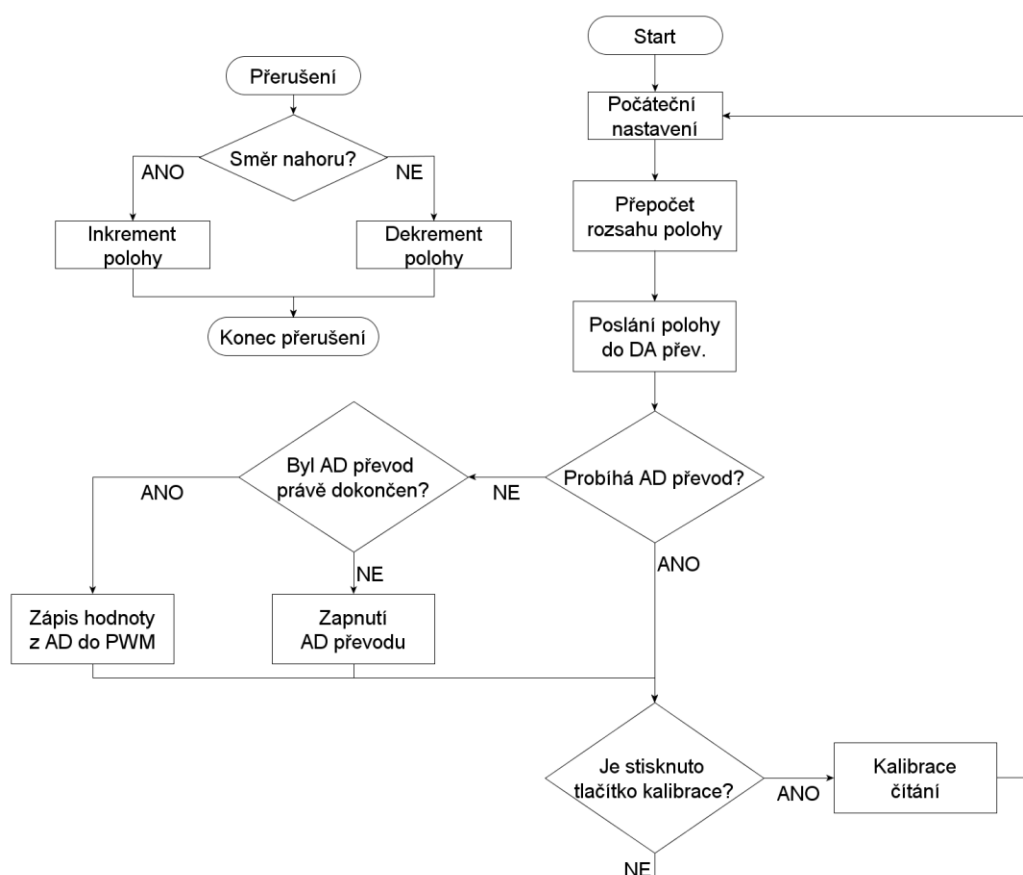
pro již invertované logické signály. Na základě výsledku je nastaven příslušný výstup udávající směr a pomocí dané funkce je na druhý výstup vyslán impulz o délce 10 μ s. Pokud by se omylem program dostal do přerušení, například vinou velmi krátkého rušivého impulzu bez změny na vstupu, nebo by nastala nějaká kombinace předchozího a současného stavu, která není v tab. 4.1, a tak není za normální funkce reálně přípustná, program vzhledem k výstupům nic neudělá. Po impulzu, případně po chybovém vyhodnocení, obsluha přerušení končí a program se opět vrací do nekonečného cyklu. Ze softwarové simulace vyplývá, že při uvedené konfiguraci a mnou použité taktovací frekvenci mikropočítače 20 MHz trvá celé vyhodnocení jedné hrany signálu maximálně



25 μ s. Budeme-li tedy pro větší spolehlivost počítat s minimálním odstupem hran signálu 30 μ s, může použitý interpolátor zpracovávat signál o maximální frekvenci 8,3 kHz, přičemž se tím rozumí, že signál o této frekvenci je přítomný na obou vstupech od čidla zároveň a za trvání jedné periody signálu na jednom vstupu tedy počítá obvod celkem 4 hrany.

4.5 Program hlavního mikropočítače

Dále bylo třeba vytvořit program pro hlavní mikropočítač kanálu. Základní struktura tohoto programu je zachycena vývojovým diagramem na obr. 4.5. Program po resetu, což je nejčastěji sepnutí napájecího napětí, zavolá funkci pro počáteční nastavení mikropočítače. Poté se program dostane do nekonečné smyčky a zároveň je povoleno asynchronní přerušení od náběžné hrany vstupu, na který je připojen impulzní



Obr. 4.5 Vývojový diagram programu hlavního mikropočítače kanálu



signál o změně polohy z interpolátoru. Dojde-li k vyvolání přerušení, zavolá se pro jeho obsluhu funkce, v níž se provede prosté přičtení případně odečtení dle informace o směru změny polohy z interpolace. Poté obsluha přerušení končí a běh programu se vrací do nekonečné smyčky. V ní se nejprve provede funkce realizující přepočet hodnoty polohy vzniklé čítáním na rozsah odpovídající D/A převodníku, tedy 0 - 4095. Přepočet se provede jednoduchým násobením konstantou, která pro kanál elevace je 7,51 a pro azimut 2,75. Jak jsem již dříve uvedl, programy pro hlavní mikropočítače obou kanálů jsou totožné pouze s výjimkou těchto dvou konstant. Ty jsou pro přehlednost uvedeny na začátku kódu a výběr správné konstanty se před přeložením programu provede pouze zakomentováním jednoho řádku dle návodu v popisku. Tyto konstanty jsem použil s přesností pouze na dvě desetinná místa, neboť počítání s větší přesností způsobovalo nefunkčnost celého programu. Patrně jsem zde narazil na nějaký problém, který výrobce překladače neošetřil. Podrobnější příčinu se mi zjistit nepodařilo, neboť v počítačové simulaci program fungoval korektně, ale naopak v součástce prokazatelně kvůli tomuto selhával. Vráťm-li se k funkci přepočtu rozsahu, zbývá o ní jen dodat, že pokud by došlo k nestandardnímu načítání impulzů v krajní poloze, tak jak to bylo zmíněno v kapitole 2, čítač je v přerušení připočte. Poté při zpětném pohybu odečte, takže nedojde ke ztrátě kalibrace. Při přepočtu hodnoty se tyto impulzy navíc neprojeví, místo nich se použije pevné minimum nebo maximum.

Hodnota polohy přepočtená dle předchozího popisu je dále pomocí speciální funkce zadána do modulu pro obsluhu sériové sběrnice SPI, který ji pošle do D/A převodníku spolu s potřebným nastavením pro tento převodník jako 16bitové slovo. Přenos je dle mého nastavení taktován frekvencí 5 MHz, díky čemuž trvá přenos celé informace 3,2 μ s. Z toho vidíme, že zpoždění způsobené sériovou komunikací je vzhledem k dané fyzikální aplikaci zanedbatelné. Smyčka programu poté testuje, zda je v běhu A/D převod, nebo zda došlo k jeho skončení. A/D převodník totiž pracuje po spuštění převodu nezávisle na probíhajícím kódu. Pokud A/D převod neběží a není aktuálně po jeho dokončení, převod je spuštěn. Naopak, běží-li převod a není ještě dokončen, program pokračuje dál a neprovede vzhledem k A/D převodníku a PWM modulu žádnou akci. V případě, že je převod dokončen, zavolá se funkce, která vyčte data z A/D převodníku a zapíše je do PWM modulu. Celý převod trvá podle [7] 9 časovacích cyklů, které mají v mém nastavení periodu 3,2 μ s, doba úplného A/D převodu



je tedy 28,8 μ s. PWM modul poté na základě zapsané hodnoty upraví střidu výstupního PWM signálu. Tato střída se vždy mění na konci jedné periody výstupního signálu, přičemž jeho frekvence je nastavena dle [7] na hodnotu 19,53 kHz, což znamená periodu 51,2 μ s.

Po zapsání hodnoty do PWM modulu program testuje, zda je stisknuté tlačítko pro kalibraci odměřování. Není-li tomu tak, program ukončí tuto iteraci smyčky a dostane se na začátek další. Pokud však je tlačítko stisknuté, zavolá funkci provádějící kalibraci tak, že nastavuje hodnotu v proměnné, ve které se ukládá aktuální hodnota polohy. Nastavena je zde hodnota rovna polovině rozsahu. To je z důvodu, že model vrtulníku má aretaci os umístěnou právě v polovinách rozsahů pohybů. Nakonec tato funkce zhasne signalizační diodu, informující uživatele o nutnosti kalibrovat odměřování. Poté program zahájí další běh smyčky. Zmíněná dioda je rozsvícena vždy po resetu programu mikropočítače. Po resetu je též provedeno načtení kalibrační hodnoty do proměnné pro polohu. Nicméně vzhledem k tomu, že není možné se vždy spolehnout na to, že uživatel modelu při zapnutí napájení bude mít aretovány obě osy, je popsanou signalizací nucen toto zkontrolovat a má tak jistotu o přesné kalibraci. Jakékoliv ukládání polohy před vypnutím systému a její opětovné načtení nemá v této aplikaci význam díky použití inkrementálních čidel a nemožnosti zamezení pohybu v osách ovládacím systémem.



5 Výroba a zabudování ovládacího systému

5.1 Návrh a výroba plošného spoje

Pro kreslení schémat a následnému návrhu plošného spoje jsem použil software EAGLE. Výsledný plošný spoj, jehož motiv jsem umístil do přílohy D, je jednostranný a obsahuje nutné drátové propojky, které jsou zakresleny jako vodiče v horní vodivé vrstvě. Vzhledem k tomu, že plošný spoj jsem následně osazoval a pájel ručně a nebyly v této aplikaci kladeny velké požadavky na miniaturizaci zařízení, jsou všechny součástky typu THT (Through-hole Technology), což znamená, že prvky mají drátové vývody, které vedou po osazení skrz plošný spoj. Všechny integrované obvody, s výjimkou reference a stabilizátorů, jsem umístil do precizních patič. V zapojení je několik digitálních a analogových signálů. Žel, není možné použité zapojení striktně rozdělit na analogovou a digitální část. To je způsobeno tím, že některé integrované obvody zpracovávají zároveň oba typy signálů. Jde o hlavní mikrokontroléry a D/A převodníky. Oba obvody mají též pouze jeden vývod pro zemnění, takže ani z tohoto důvodu nelze uplatnit přísná pravidla pro oddělování analogové a digitální části obvodu. Přesto všechno jsem se snažil, aby část, která je převážně analogová, byla topologicky umístěna v jedné části desky a převážně digitální část byla na druhé straně.

Při návrhu rozložení součástek jsem jako prioritu měl co nejkratší vedení rychlých digitálních signálů, zejména sběrnice SPI. Dále jsem se snažil o co nejkratší vedení hodinového signálu od krystalů jednotlivých mikrokontrolérů. Všechny ostatní signály nejsou tak rychlé, a tak nejsou ani tolik citlivé na délku vedení. Obzvláště u analogových signálů nejsou očekávané rychlé průběhy, tudíž na navržené desce jejich délka nemohla být kriticky dlouhá. Nicméně jsem se samozřejmě snažil mít jejich vedení co nejkratší v rámci možností. Pro omezení rušení stíněním a pro lepší propojení země jsem použil tzv. rozlévanou měď. To znamená, že spoji nevyužitá místa na desce je dodatečně pokryto vodivou plochou, která je v mém případě galvanicky spojena se zemněním. Především jsem dbal, aby byly co nejvíce eliminovány proudové špičky v napájecím vedení. To jsem zajistil jejich blokováním, umístěním blokovacích kondenzátorů ke každému integrovanému obvodu. Dále jsem dbal na co nejkratší propojení země napájení integrovaného obvodu a uzemnění časovacího obvodu. Ne vždy to bylo snadné díky nevhodnému rozmístění napájecích pinů u jednotlivých součástek



a v několika případech jsem musel kvůli zkrácení zemnicího vedení některé méně důležité signály vést drátovou propojkou.

Plošný spoj jsem vyráběl tzv. fotocestou ručně dle následujícího popisu. Nejprve jsem přes folii s motivem spoje osvětlil desku s fotocitlivou vrstvou. Osvěcoval jsem pomocí UV lampy. Dále ve 2% roztoku hydroxidu sodného jsem odstranil osvětlenou část fotocitlivé vrstvy a tím odkryl měď určenou k odleptání. Plošný spoj jsem poté leptal ve vodném roztoku peroxodisíranu sodného. Následně jsem odstranil pomocí etanolu zbytek fotocitlivé vrstvy, plošný spoj nalakoval pájecím a ochranným lakem. Nakonec jsem vyvrtal všechny otvory.

5.2 Osazení a oživení desky plošných spojů

Osazení jsem provedl dle osazovacího plánu umístěného v příloze D. Nejprve jsem na desku umístil všechny drátové propojky a patice pro integrované obvody. Poté jsem osadil konektory a tlačítka. Dále jsem umístil všechny diskrétní součástky počínaje pasivními až po aktivní. Krystaly bylo nutno přichytit a propojit jejich pouzdra pomocí drátků se zemí. Nakonec jsem osadil velké součástky jako filtrační kondenzátor a stabilizátory s chladiči. Po dokončení pájení jsem vizuálně zkontroloval všechny spoje, zda nedošlo někde v rozměrově kritických místech k nežádoucímu propojení pájkou.

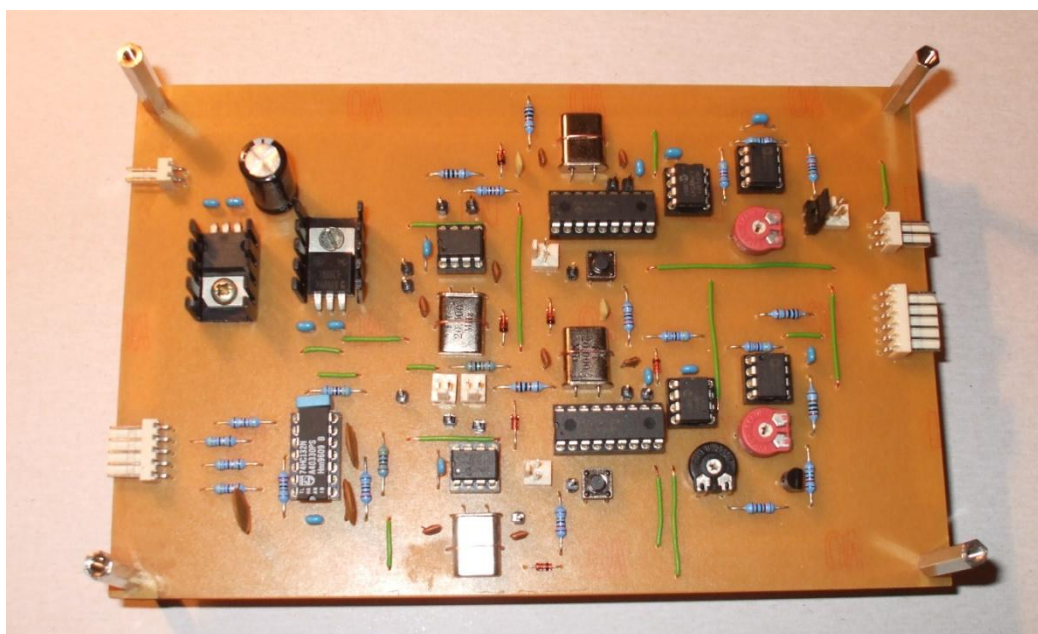
Oživení jsem provedl podle následujícího postupu. Nejprve jsem připojil vstupní napájecí napětí 12 V na desku. Zkontroloval jsem, zda jsou obě napájecí napětí jak 5 V tak i 10 V přítomny na správných pinech patic. Poté jsem trimrem R5 nastavil přesně referenční napětí 5V pro D/A převodník. Následně jsem umístil do patic oba operační zesilovače IC9 a IC10, přičemž jsem při každém zasouvání integrovaných obvodů měl vypnuté napájení. Zkratoval jsem a uzemnil oba vstupy operačních zesilovačů a pomocí trimrů R10 a R18 jsem nastavil kompenzaci vstupní napěťové nesymetrie tak, aby výstupní napětí v tomto stavu bylo co nejmenší. Dále jsem osadil do patic D/A převodníky IC3 a IC8, všechny mikrokontroléry IC4, IC5, IC6 a IC7.

Poté jsem do těchto mikrokontrolérů zavedl příslušné programy. To jsem provedl pomocí programovacího zařízení s názvem Presto od firmy ASIX s.r.o. Vzhledem k tomu, že může i v budoucnosti vyvstat potřeba zavést nový program, například bude-li nutné změnit rozsah měření, nebo pokud bude třeba doplnit zapojení o další funkci, uvádím zde podrobný postup. Nejprve bylo třeba samozřejmě připojit zařízení Presto



k počítači s již nainstalovaným softwarem pro něj. Ten je dodáván se zařízením. Následně jsem propojil jeho výstupy s programovacími piny pro daný mikropočítač. Tyto piny jsou ve všech schématech umístěných v příloze B zřetelně označeny. Dle plánu rozmístění součástek v příloze D jsem našel umístění těchto pinů na desce plošných spojů. Nejdříve jsem propojil zemnicí vývod označený GND a napájení 5 V, na zařízení Presto označené symbolem VDD. Následně jsem vývod VPP připojil na pin MCLR, což znamená master clear, daného mikropočítače. Poté již stačilo propojit vývody DATA a CLOCK s podobně nazvanými piny PGD a PGC. Pak již bylo možné do daného mikropočítače zavést program v podobě souboru s příponou .hex. Ten je výsledkem překladu v prostředí MPLAB. Toto se dělo při použití napájecího napětí z desky ovládacího systému. Po zavedení všech programů jsem ještě do patice umístil integrovaný obvod IC1. Kompletní hotová deska je vidět na obr. 5.1.

Nakonec bylo třeba vyzkoušet měřeními všechny základní funkce systému. Při tomto měření jsem pouze ověřoval funkčnost celého zapojení, podrobnější měření následuje v kapitole 6. Nejdříve jsem ověřil převod signálů pro ovládání motorů na výstupní PWM signál. Poté jsem vyzkoušel převod ze vstupů od IRC čidel na analogové výstupy. Zároveň jsem prověřil funkci povolování motorů v součinnosti s magnetickým kontaktem. Nakonec jsem ještě ověřil funkci kalibrace odměřování úhlu natočení. Funkci indikační LED jsem ověřil opticky po resetování běhu programů mikrokontrolérů IC6 a IC7.

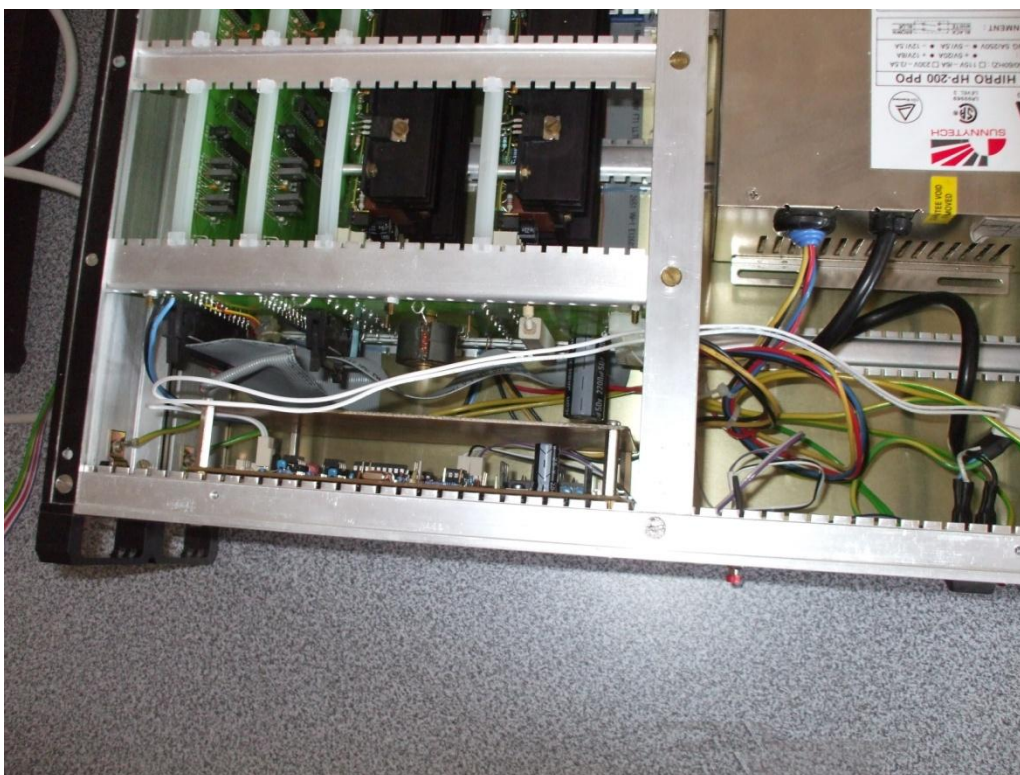


Obr. 5.1 Foto hotové desky plošných spojů



5.3 Zabudování ovládacího systému

Vyrobenou desku s ovládacím systémem jsem nainstaloval do přístrojové krabice se zdrojem a výkonovými budiči pro motory. Provedl jsem to způsobem, že jsem ji připevnil pomocí čtyř distančních sloupků k čelnímu panelu. Mezi desku a ostatní části zařízení jsem také na distanční sloupky umístil stínění v podobě cuprexitové desky s jednou vrstvou mědi pokrývající celou plochu. Výsledek je vidět na obr. 5.2, kde deska ovládacího systému je nejblíže čelnímu panelu (na obr. 5.2 dole). Dále jsem na čelní panel umístil LED signalizující nutnost kalibrace odměřování a tlačítko provádějící kalibraci.



Obr. 5.2 Foto zabudovaného ovládacího systému

Dále bylo nutné provést elektrické zapojení ovládacího systému. To jsem provedl dle schématu umístěného v příloze E. Nejdříve jsem připojil zmíněné tlačítko a signalizační LED diodu. Poté jsem připojil magnetický kontakt, který jsem umístil do krabice pro budoucí použití. Tento kontakt má v budoucnosti spínat povolovací signál pro motory v závislosti na přítomnosti ochranné klece. Vzhledem k tomu, že v době dokončení jsem neznal přesné umístění modelu a klece, magnetický kontakt jsem vyřadil



z provozu propojením pinů 1 a 2 pinového pole JP19. Umístění tohoto pole je vidět v příloze D. Pro uvedení magnetického kontaktu do funkce stačí přesunout propojku na JP19 tak, aby propojovala piny 2 a 3.

Následně jsem odstranil z krabice starý výstupní konektor s kabelem a nahradil jsem ho stejným novým pouze s jiným zapojením. Jeho propojení je vidět jednak na schématu v příloze E a také je zapojení pinů tohoto konektoru uvedeno v tab. 5.1. Tato tabulka má zásadní význam pro další použití modelu. Podotknul bych ještě, že ostatní vývody konektoru, které nejsou v tab. 5.1 uvedeny, nebyly použity. Dle výše zmíněného schématu jsem také připojil všechny zbývající signály a napájení k vnitřní propojovací desce, která je vidět na obr. 5.2, přičemž je to nejbližší rovnoběžná deska s ovládacím systémem, nepočítám-li stínící desku. Připojení jsem provedl tak, že na základě zapojení starého výstupního konektoru jsem našel spoje s těmito signály a k nim jsem připájel vodiče.

Tab. 5.1 Zapojení vývodů výstupního 37pinového konektoru

číslo vývodu	typ	funkce
3	vstup	ovládání hlavního motoru napětím 0 – 10 V
5	vstup	ovládání postranního motoru napětím 0 – 10 V
7	výstup	poloha na v ose elevace napětí 0 – 10 V
9	výstup	poloha na v ose azimut napětí 0 – 10 V
10		GND
11		GND
29		GND
35	vstup	povolení běhu motorů - ENABLE - TTL úrovně, povoleno při log. 1



6 Ověření funkčnosti ovládacího systému

6.1 Měření opakované přesnosti odměřování úhlů natočení

Nejprve jsem zkoumal opakovanou přesnost odměřování. To jsem provedl tak, že jsem na začátku měření v aretované poloze osy kalibroval odměřování. Poté jsem několikrát uvolněným vrtulníkem velmi rychle pohyboval. Tento pohyb zahrnoval i prudké nárazy v krajní poloze. Následně jsem vrtulník opět aretoval a změřil výstupní napětí odpovídající poloze v příslušné ose. Potom jsem opakoval pohyb vrtulníkem a měření napětí odpovídající poloze při aretaci. Pro každou osu jsem provedl měření o deseti náměrech. Výsledky tohoto měření jsou uvedeny v tab. 6.1, kde n představuje pořadí náměru a U výstupní napětí odpovídající poloze. Z výsledku jasně vyplývá, že v ose elevace je odměřování přesné a spolehlivé. V hodnotách pro osu azimut je vidět rozdíl mezi jednotlivými náměry. Ten však není způsoben chybou ovládacího systému, ale špatnou aretací. Otvor pro kónický aretační šroub je již značně opotřebován. A tak i při maximálním utažení šroub nedosedne vždy na totožné místo a lze s modelem dokonce v malém rozsahu pohybovat. Tento rozsah ve výstupním napětí byl od velikosti 4,918 V do 5,006 V.

Tab. 6.1 Hodnoty z měření opakované přesnosti

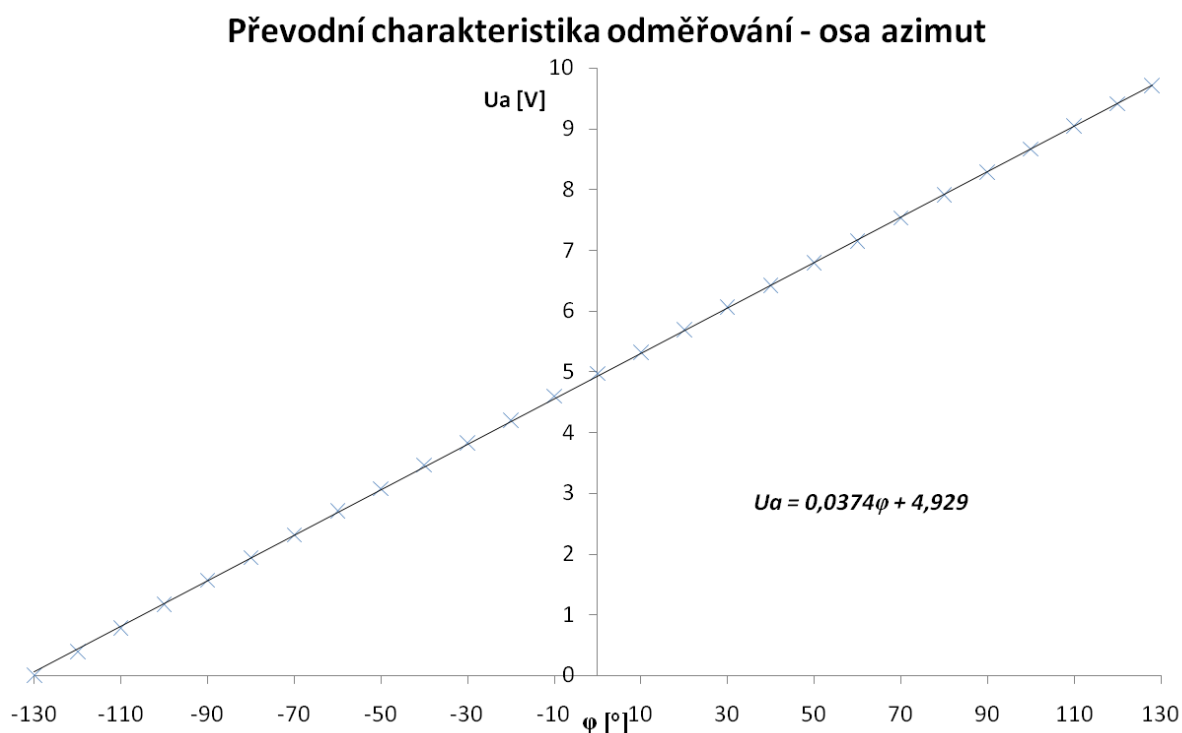
Azimut				Elevace			
n	$U [V]$	n	$U [V]$	n	$U [V]$	n	$U [V]$
1	4,968	6	4,967	1	4,971	6	4,971
2	4,973	7	4,973	2	4,971	7	4,971
3	4,968	8	4,973	3	4,971	8	4,971
4	4,973	9	4,973	4	4,971	9	4,971
5	4,973	10	4,973	5	4,971	10	4,971

Voltmetr: Pro'sKit MT-1860 v.č. 992958921



6.2 Měření převodních charakteristik odměřování úhlů natočení

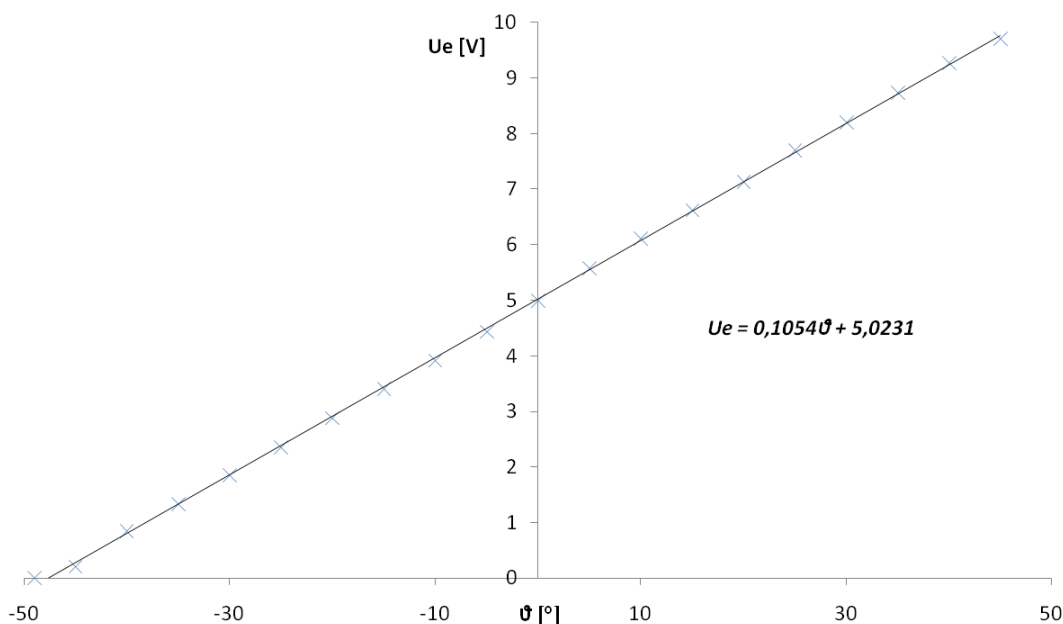
Změřil jsem dále převodní charakteristiku odměřování, tedy závislost výstupního napětí na úhlu natočení vrtulníku v příslušné ose. Konvence označení os je popsána v kapitole 1. Kladné smysly pohybu v osách jsou vyznačeny na obr. 1.2. Úhel jsem odečítal na rýsovacím úhloměru vhodně připevněném k modelu. Hodnoty z tohoto měření zde pro přehlednost neuvádím, jsou umístěny spolu se soupisem přístrojů v příloze F. Jako výsledek tohoto měření pro osu azimut je na obr. 6.1 výsledná charakteristika i s rovnicí získanou lineární regresí. Přičemž φ představuje úhel natočení v dané ose a U_a výstupní napětí jemu odpovídající. Hodnota úhlu $\varphi = 0^\circ$ odpovídá poloze při aretaci. Tvar výsledné charakteristiky je lineární s prakticky nepozorovatelnou odchylkou v jednotlivých bodech.





Podobný výsledek má i měření převodní charakteristiky odměřování v ose elevace. Ten je v podobě grafu na obr. 6.2. Proměnná ϑ zde představuje úhel natočení v ose a U_e výstupní napětí mu odpovídající. Hodnota úhlu $\vartheta = 0^\circ$ odpovídá poloze při aretaci. Opět je převod lineární, nicméně je zde drobná odchylka v počátku charakteristiky. Ta je patrně způsobena malým pásmem necitlivosti, které je v krajní poloze, a špatným odečtením úhlu.

Převodní charakteristika odměřování - osa elevace



Obr. 6.2 Převod odměřování v ose elevace s rovnicí z regrese

Tímto měřením jsem ověřil linearitu převodu signálů z IRC čidel na výstupní napětí. Rovnice získané lineární regresei obou charakteristik mají velký význam pro další použití modelu. Po malé úpravě je možné na jejich základě převádět unifikovaný napěťový signál na úhel natočení v dané ose. Vztah pro výpočet úhlu natočení v ose azimut tedy je:

$$\varphi = \frac{U_a - 4,929}{0,0374} [^\circ], \quad (6.1)$$

kde U_a [V] představuje výstupní napětí pro tuto osu. Obdobně úhel v ose elevace lze vypočítat podle vztahu:

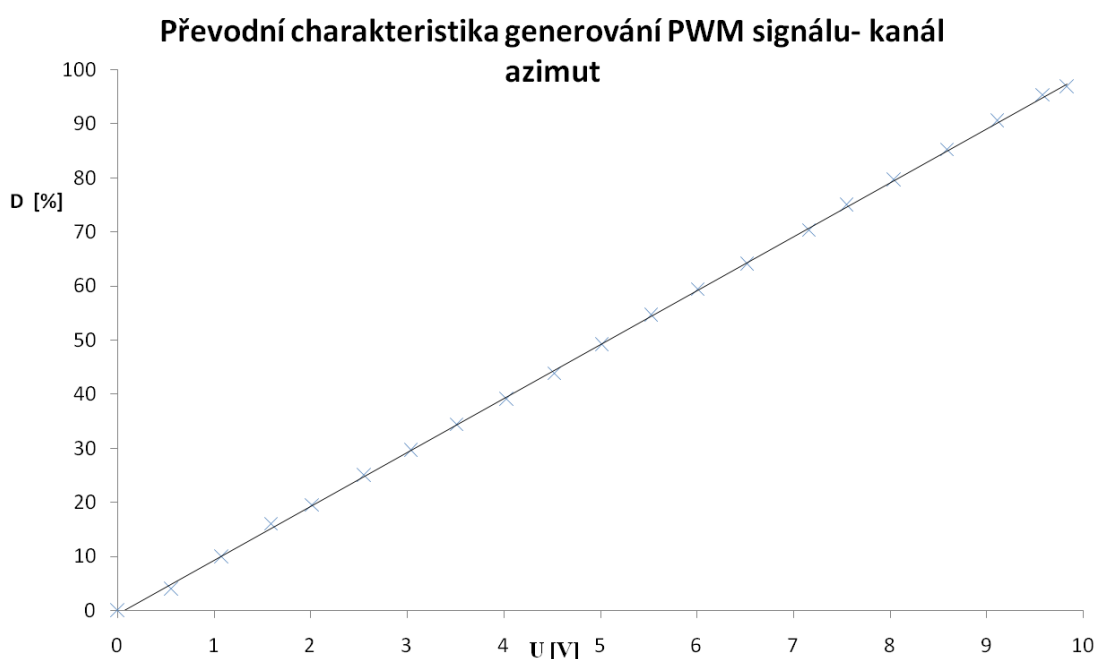
$$\vartheta = \frac{U_e - 5,0231}{0,1054} [^\circ], \quad (6.2)$$

přičemž U_e [V] zde analogicky je výstupní napětí této osy.



6.3 Měření charakteristiky převodu napětí na signál PWM

V tomto měření jsem změřil závislost střidy výstupního signálu PWM pro budiče motorů na vstupním ovládacím napětí. Hodnoty získané tímto měřením, stejně jako použité přístroje, zde opět pro přehlednost neuvádím. Jsou umístěny v příloze F. Jako výsledek je na obr. 6.3 převodní charakteristika. Zde D představuje střidu signálu v procentech a U_{IN} vstupní napětí. Výsledek pro kanál elevace je naprosto totožný, a tak



Obr. 6.3 Graf převodu napětí na střidu PWM signálu – kanál azimut

jej zde neuvádím. Z grafu je vidět, že převod mezi vstupním napětím a střidou výstupního signálu je lineární. Jsou zde v jednotlivých náměrech drobné odchylky, které jsou patrně způsobeny nepřesností automatického měření střidy v osciloskopu. Při měření nižších hodnot se údaj o střidě poměrně značně měnil za konstantního nastavení osciloskopu i vstupního napětí.



7 Možnosti využití modelu a ovládacího systému

Model v součinnosti s ovládacím systémem má samozřejmě hlavní uplatnění ve výuce předmětů zabývajících se automatickým řízením. Vrtulník představuje nelineární nestabilní systém se dvěma vstupy a dvěma výstupy. Mezi jednotlivými proměnnými je silná interakce a dynamické chování modelu je velmi rychlé. To je dáno fyzikální podstatou vrtulníku. Typickou úlohou spojenou s tímto modelem může být návrh optimálního regulátoru. Jistě je možné provádět identifikaci soustavy v různých pracovních bodech. Dále mohou být úlohy zadávané na tomto modelu různě modifikovány. Například je možné řešit řízení vrtulníku při aretaci jedné osy, čímž se řízení i jeho návrh zjednoduší. Naopak je možné úlohy zkomplikovat a učinit je tak zajímavějšími například přidáním závaží na vhodné místo ve vrtulníku.

Vrtulník je dále možné využít k testování pokročilejších odměřovacích a řídicích systémů. Nabízí se například umístit na model gyroskopické snímače informující o úhlových rychlostech pohybu v jednotlivých osách. Jednak je možné zaměřit se pouze na konstrukci takovýchto snímačů. Nebo i na navazující obvody, které by vypočítávaly polohu. U takového měření by bylo užitečné prozkoumat opakovanou absolutní přesnost díky tomu, že je vždy k dispozici relativně přesný údaj o poloze. Je možné jít ještě dál a přenášet data z takovýchto snímačů k řídicímu systému bezdrátově. Vzhledem k tomu, že model má rychlou dynamiku, může být užitečné prozkoumat spolehlivost přenosu rychlých informací různými bezdrátovými technologiemi.

Je také možné po úpravě využít samotný návrh realizovaného ovládacího systému pro jiné podobné modely a soustavy obsahující inkrementální čidla a stejnosměrné motory. Jistě přichází v úvahu využití částí mého řešení ovládacího systému. Například řešení čtyřnásobné interpolace pomocí jednočipového mikropočítače o osmi vývodech je velmi efektivní a levné. Lze jej použít v mnoha aplikacích s inkrementálními čidly jak rotačními tak i lineárními. Přičemž, jak jsem uvedl v kapitole 4.4, může takový interpolátor spolehlivě vyhodnocovat signály o frekvenci až 8,3 kHz. Dále by bylo navíc možné tuto frekvenci ještě zvýšit, pokud by postačoval výstupní impuls kratší než 10 μ s.



8 Závěr

Dle zadání jsem prozkoumal stav ovládacího systému modelu vrtulníku HUMUSOFT CE150. Dále jsem diskutoval několik možností převodu signálů z polohových čidel na výstupní napěťový signál 0 – 10 V. Též jsem uvažoval možnosti generování spínacího signálu pulzně šířkové modulace v závislosti na vstupním napětí 0 – 10 V. Na základě zjištěných skutečností a porovnání jednotlivých variant jsem navrhl mé řešení ovládacího systému. Toto řešení je založeno především na použití jednočipových mikropočítačů s periferními obvody. Navržený ovládací systém jsem po úspěšném odzkoušení formou prototypu zkonstruoval jako zařízení na jedné desce plošných spojů. Systém jsem následně zabudoval do přístrojové krabice zdroje a budičů. Díky tomu k modelu nepřibylo žádné externí zařízení, které by ztěžovalo manipulaci s ním.

Měřením jsem otestoval funkčnost zvoleného řešení. Ovládací systém v interakci s modelem vrtulníku funguje spolehlivě. Odměřování poskytuje přesnou informaci o aktuální poloze vrtulníku. Realizovaný systém nezanesl do celé soustavy žádnou další nelinearitu. Model vrtulníku je nyní možno využívat při výuce v součinnosti s univerzální kartou pro software Matlab nebo s průmyslovým regulátorem. Použití unifikovaných vstupů a výstupů umožňuje oproti systému od výrobce modelu takovouto flexibilitu. Finanční náklady na praktickou realizaci zvoleného ovládacího systému nepřesáhly 2000,- Kč, což je necelá desetina ceny systému, který nabízí výrobce modelu. Domnívám se, že zadání mé práce jsem beze zbytku splnil.



Použitá literatura

- [1] *Helicopter Technical Manual*. ČR : HUMUSOFT s.r.o, 2002. 20 s.
- [2] *LS7083/7084* [online]. [s.l.] : LSI Computer Systems, Inc., 2000 [cit. 2011-05-01]. Dostupné z WWW: <www.alldatasheet.com>.
- [3] ROBENEK, Jan. *Www.hw.cz* [online]. 2010 [cit. 2011-05-10]. TimerBlox – 5 obvodů, 5 základních funkcí a „555“ různých aplikací. Dostupné z WWW: <<http://hw.cz/novinky/art3722-timerblox-5-obvodu-5-zakladnich-funkci-a-555-ruznych-aplikaci.html>>.
- [4] Redakce HW serveru. *Www.hw.cz* [online]. 2001 [cit. 2011-05-10]. Efektivně na PWM s x51. Dostupné z WWW: <<http://hw.cz/Teorie-a-praxe/Programovani/ART471-Efektivne-na-PWM-s-x51.html>>.
- [5] Microchip, Inc. *MCP4921/4922* [online]. 2004 [cit. 2011-05-16]. Dostupné z WWW: <www.alldatasheet.com>.
- [6] ANALOG DEVICES, Inc. *AD820* [online]. 2004 [cit. 2011-05-10]. Dostupné z WWW: <www.alldatasheet.com>.
- [7] Microchip, Inc. *PIC16F87/88 Data Sheet* [online]. 2003 [cit. 2011-05-11]. Dostupné z WWW: <www.alldatasheet.com>.



Příloha A - Obsah přiloženého CD

BakPraceJiriBudasz.docx - Text práce ve formátu docx.

BakPraceJiriBudasz.pdf - Text práce ve formátu pdf.

OvlSystemVrtulniku-bPrace - Projekt pro návrh a zabudování desky v programu EAGLE obsahující všechna schémata a výkres desky plošných spojů.

ProgramHlavniMikropoc.c - Kód programu pro hlavní mikropočítače v jazyce C.

ProgramIRC.c - Kód programu pro interpolační mikropočítač v jazyce C.

HlMikropAZIMUT.hex - Přeložený program pro hlavní mikropočítač kanálu azimut.

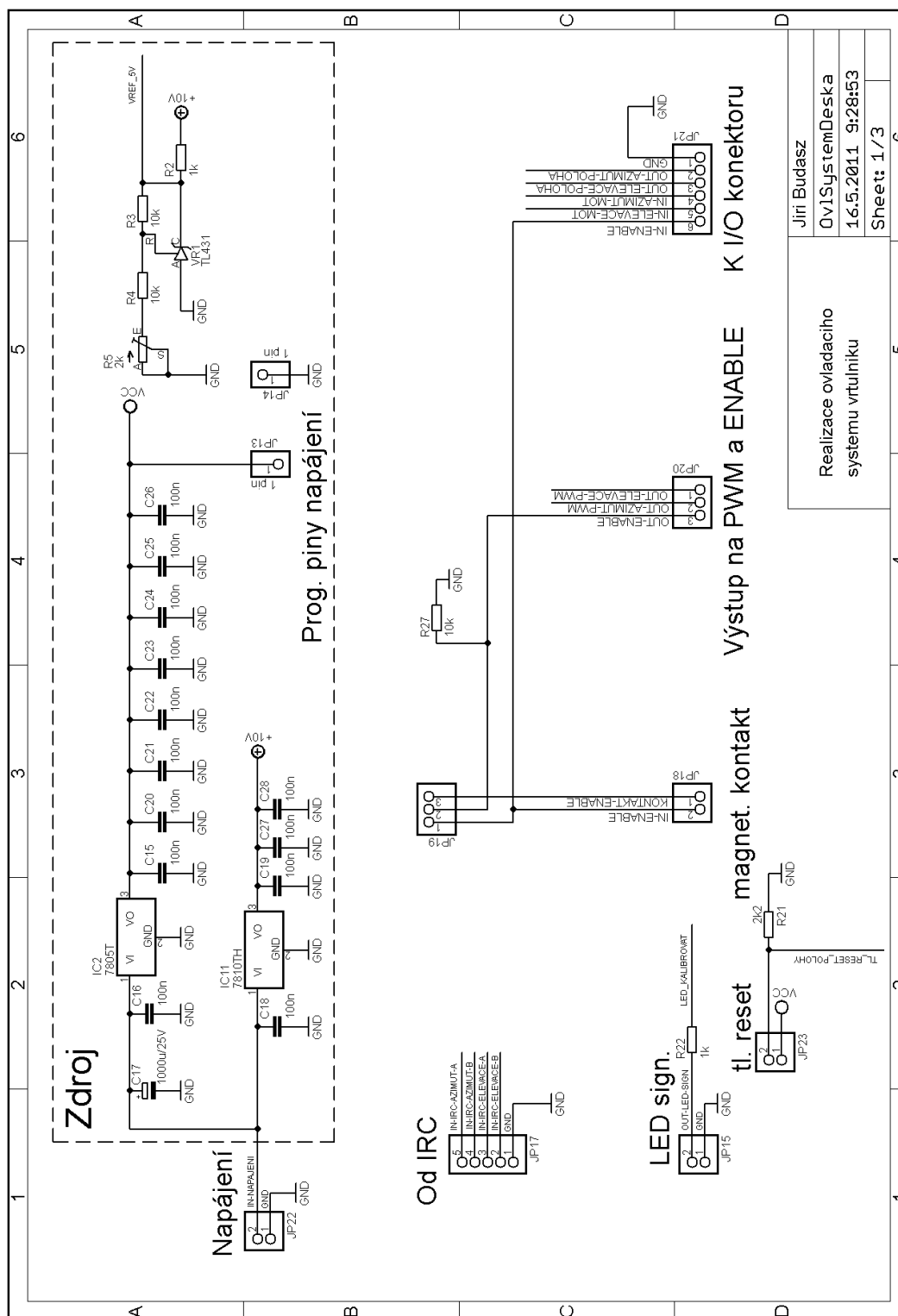
HlMikropELEVACE.hex - Přeložený program pro hlavní mikropočítač kanálu elevace.

InterpolatorIRC.hex - Přeložený program pro interpolační mikropočítač.

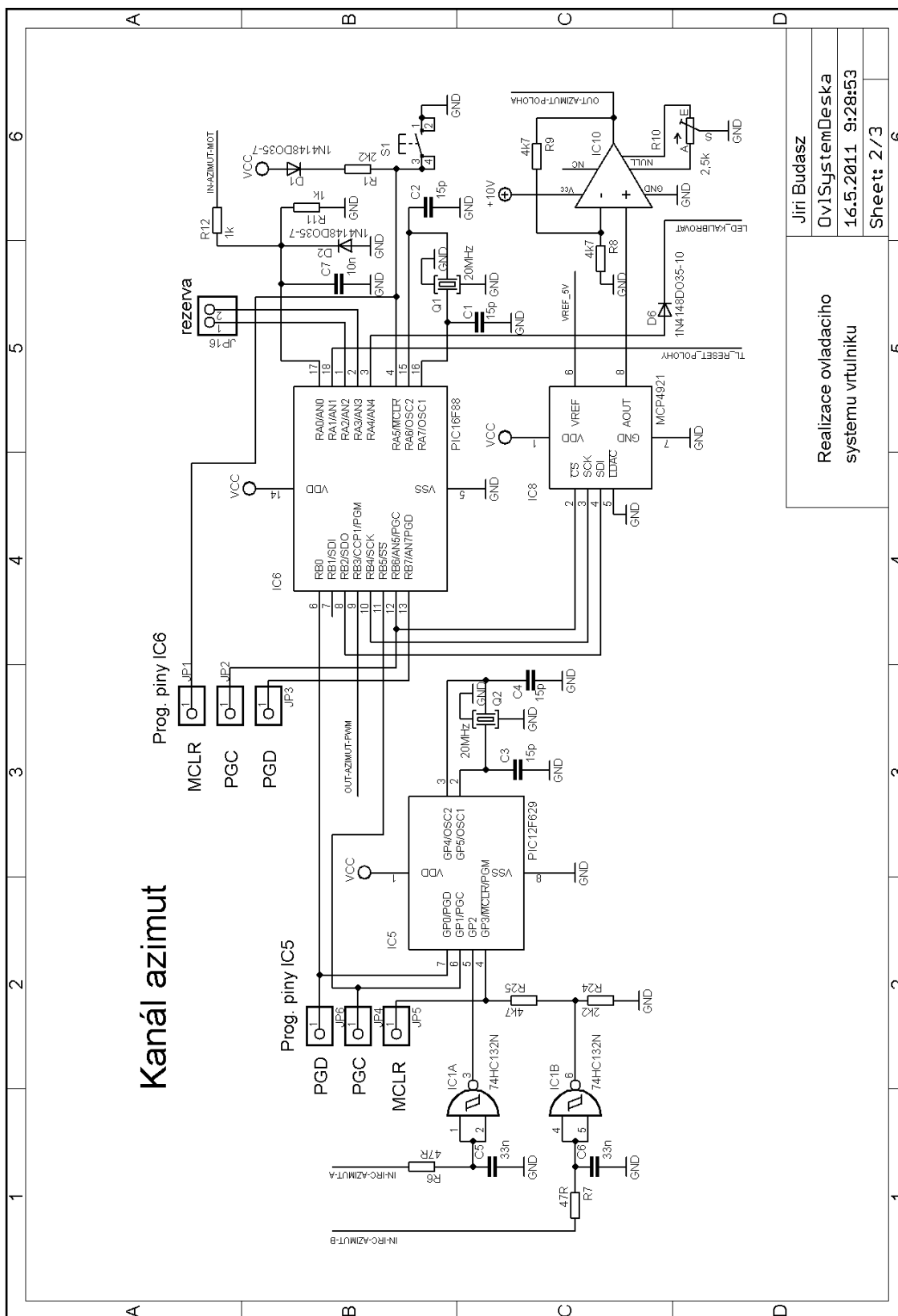


Příloha B - Schémata zapojení ovládacího systému

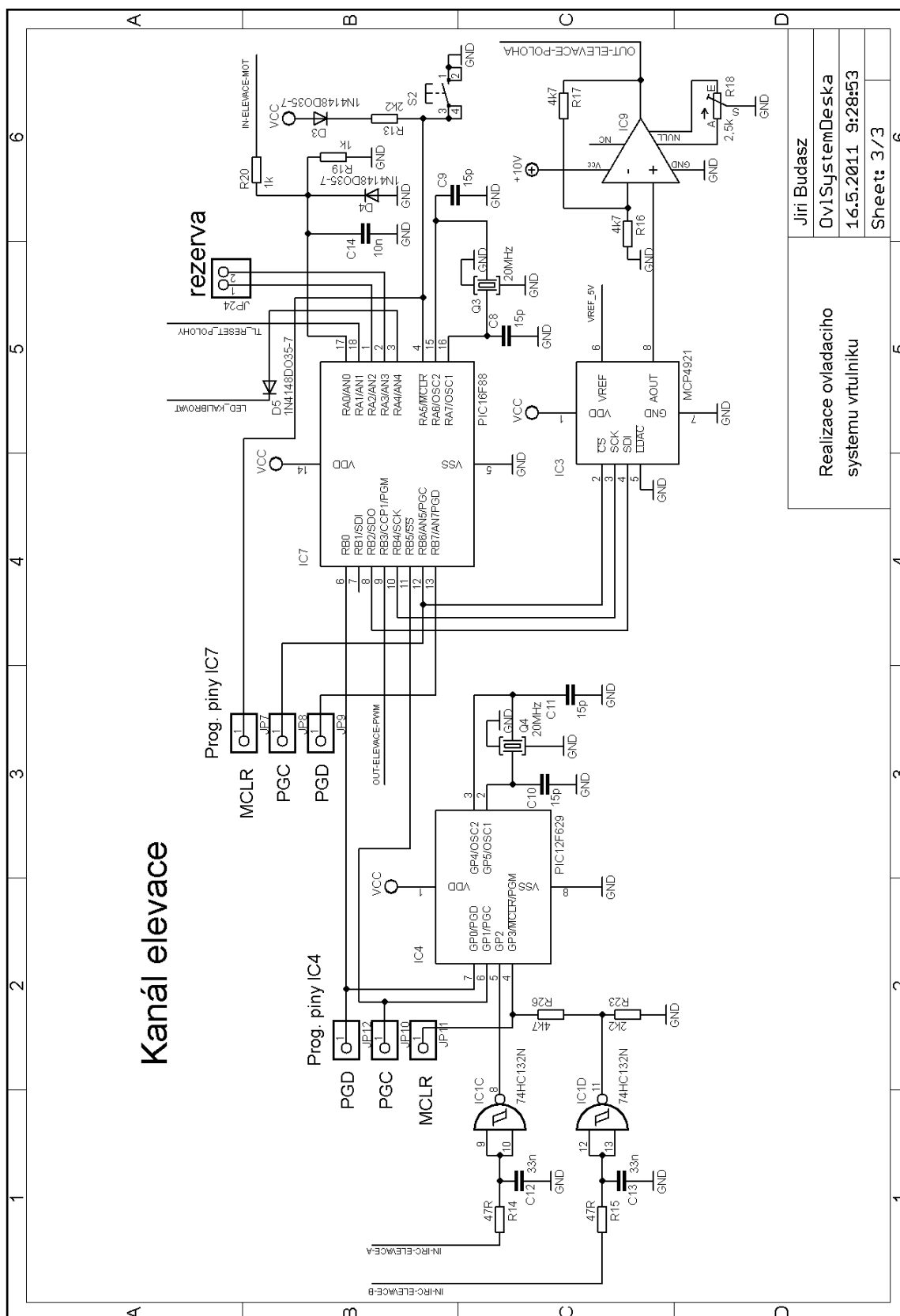
Část zdrojů a konektorů



Kanál azimut



Kanál elevace





Příloha C - Výpisy programů

Program mikropočítače pro čtyřnásobnou interpolaci

```
// .....
//      Jiri Budasz
//      listopad 2010 - duben 2011
//      Program pro mikropocitac PIC12F629 pro 4nasobnou interpolaci
//      signalu IRC cidel
//      verze 1.0
//      Reseni bakalarske prace pro RSS na FM TUL
//      Vstupy:      GP2 - vstup IRC A - negovany
//                  GP3 - vstup IRC B - negovany
//
//      Vystupy:     GP0 - vystup kladne impulzy o zmene polohy
//                  GP1 - vystup smer zmeny polohy
// .....
#include <htc.h>
#include <math.h>
#include <stdio.h>
#define _XTAL_FREQ 20000000
// .....
//      Nastaveni konfiguracniho slova uP
//      taktovani krystalem do 20MHz, watchdog vypnut,
//      MCLR vypnut GP3 jako dig vstup,
//      reset po zapnuti PWRT zapnut,
//      reset poklesem napajeciho napeti BOR vypnut
// .....
CONFIG(HS & WDTDIS & MCLRDIS & PWRTEN & BORDIS);
// .....
//      Deklarace globalnich promennych
// .....
unsigned char predchStav;
unsigned char stav;
// .....
//      Prototypy pouzitych funkcii
// .....
void nastaveni();
void interrupt obsluhaPreruseni();
void posliImpulzNaGPIO0();
// .....
//      Hlavni funkce vykonavana po resetu
// .....
void main(void)
{
    nastaveni();                //volani funkce pro pocatecni inicializaci
    while(1)                    //nekonecny cyklus, ceka jen na preruseni
    {
        CLRWDT();              //nulovani watchdog casovace
    }
}
// .....
//      Funkce pro vykonani pocatecniho nastaveni uP
// .....
void nastaveni()
{
    //Nastaveni portu
    TRISIO = 0b1100;                //GP0,1 vystup, GP2,3 vstup
    CMCON = CMCON | 0b111;          //odpojeni vstupu komparatoru od GP0 a GP1
    //Nulovani vystupu
    GPIO0 = 0;
    GPIO1 = 0;
    //Nastaveni preruseni
    IOCB = 0b1100;                  //povoleni prerus od hrany na GPIO2 A GPIO3
    GPIE = 1;                       //povoleni preruseni od hrany na GPIO
    GPIF = 0;                       //mazani priznaku preruseni na GPIO
    //Inicializace promennych
    stav = 0;
    predchStav = ((unsigned char)GPIO>>2)&0b1100;
                                //ulozeni pocatecniho stavu vstupu
}
```



```

//co nejblize povoleni prerus, aby se
//neztracela informace
GIE = 1; //povoleni global prerus

//Nastaveni Watchdog casovace
CLRWDT(); //nulovani watchdog casovace
PSA = 1; //preddelicka je pripojena k Watchdog
casovaci
OPTION = OPTION & 0b11111000;
//delici pomer preddelicky je 1,
//cas pretečení watchdog je 18ms
CLRWDT(); //nulovani watchdog casovace
}
//.....
// Funkce pro obsluhu preruseni,
// od jakékoli hrany na vstupech GP2 a GP3
//.....
void interrupt obsluhaPreruseni()
{
    stav = (GPIO&0b1100) | predchStav;
    //vygenerovani rozhodovací promenne stav
    //kombinace soucasneho stavu vstupu
    //a stavu pred hranou

    switch (stav){ //urceni smeru citani
                  //podle minuleho a soucasneho stavu vstupu
    case (0x0d) :
        GPIO1 = 1; //smer nahoru
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x04) :
        GPIO1 = 1; //smer nahoru
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x02) :
        GPIO1 = 1; //smer nahoru
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x0b) :
        GPIO1 = 1; //smer nahoru
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x0e) :
        GPIO1 = 0; //smer dolu
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x08) :
        GPIO1 = 0; //smer dolu
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x01) :
        GPIO1 = 0; //smer dolu
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    case (0x07) :
        GPIO1 = 0; //smer dolu
        posliImpulzNaGPIO0();
        //zavola fci pro vytvoreni imp. na vystupu
        break;
    default :
        //pokud stav neni v tabulce, nestane se nic
        break;
    }
    predchStav = (stav>>2)&0b11;
    //ulozeni soucasneho stavu
    //vstupu pro pristi preruseni
    GPIF = 0; //nulovani priznaku preruseni
}

```




```

//od vstupu na GP
}
void posliImpulzNaGPIO0()
{
    GPIO0 = 1;           //zacatek impulsu na GP0
    __delay_us(10);      //doba trvani impulsu v us
    GPIO0 = 0;           //konec impulsu na GP0
}

//.....
//      Konec kodu
//.....

```

Program hlavního mikropočítače

```

//.....
//      Jiri Budasz
//      listopad 2010 - duben 2011
//      Program pro mikropocitace PIC16f88 pro ovladaci jednotku vrtulniku
//      verze 4.0
//      Reseni bakalarske prace pro RSS na FM TUL
//      Vstupy:      RB0 - impulsy (kladne) poloha
//                  RB5 - smer zmeny polohy
//                  RA0/AN0 - analog signal pro ovladani motoru
//                  RA1 - reset polohy
//                  MCLR
//      Vystupy:     RB2/SDO - SPI data do DAC
//                  RB4/SCK - SPI clock do DAC
//                  RB6      - CS pro SPI DAC
//                  RB3      - PWM signal pro budice motoru
//                  RA1      - vystup na LED sign. rozhozenou
//                  kalibraci
//      Nastaveni programu pro kanal azimut nebo elevace pred kompilaci
//      pouze zakomentovanim jednoho radku.
//.....
#include <htc.h>
#include <math.h>
#include <stdio.h>
#define _XTAL_FREQ 20000000
//.....
//      Nastaveni konstant, aby rozah IRC odpovidal
//      vystupnimu rozsahu.
//.....

#define _kanal_azimut           //      Zadani kanalu - zakomentovat tento
                                //      radek pokud kompiluji pro elevaci,
                                //      pro kanal azimut ponechat tento
                                //      radek.

#ifndef _kanal_azimut
#define _kanal_elevace
#endif
#ifdef _kanal_azimut           //      Prirazeni ciselnych konstant prepocctu
#define _konst_prevodu 2.75   //      dle zadaneho kanalu.
#define _maximum_citace 1488
#define _poc_stav_citace 744
#else
#define _kanal_elevace
#define _konst_prevodu 7.51
#define _maximum_citace 544
#define _poc_stav_citace 272
#endif
//.....
//      Nastaveni konfiguracniho slova c. 1 uP
//      vystup PWM na pinu RB3, LVP programovani vypnuto, MCLR na RA5 povolen
//      funkce resetu BOR vypnuta,
//      reset po pripojeni napajeciho napeti PWRT zapnut
//      WATCHDOG timer vypnut, cacovani krystalem do 20MHz
//.....
CONFIG( CCPRB3 & LVPDIS & MCLREN & BORDIS & PWRTEN & WDTDIS & HS );
//.....

```



```
//      Nastaveni konfiguracniho slova c. 2 uP
//      zakaz prepinani mezi vnitřním a vnějším oscilátorem
//      zakaz monitorovani takt. frekvence
//.....
CONFIG( IESODIS & FCMDIS );
//.....
//      Deklarace globalnich promennych
//.....
unsigned char kVyhozeni;
int citac;
int dataPomoc;
unsigned char vystup0;
unsigned char vystup1;
unsigned int hodnotaIN;
int poloha;
bit smer;
bit behADprev;
//.....
//      Prototypy pouzitych funkci
//.....
void nastaveni();
void posliDoDA(unsigned int);
void startADprevodu();
void zapis_AD_PWM();
void resetPolohy();
void prepocetRozsahu(int data);
//.....
//      Hlavni funkce vykonavana po resetu
//.....
void main(void)
{
    nastaveni();                //volani funkce pro pocatecni nastaveni po resetu
    while(1)                    //nekonecny cyklus
    {
        prepocetRozsahu(citac);    //zavola funkci, která prepocte polohu
                                    // na spravny rozsah
        posliDoDA(poloha);        //zavola fci, která posle akt. hodnotu polohy
                                    //do DA
        if(GODONE == 0 && behADprev == 0) //pokud nebezi AD prevod a nebyl cerstve
                                    //dokoncen
        {
            GODONE = 1;            //zahaji AD prevod
            behADprev = 1;        //nastaveni pomoc bitu
        }

        if(behADprev == 1 && GODONE == 0) //pokud skoncil AD prevod
        {
            zapis_AD_PWM();        //zavola funkci, která zapise hodnotu
                                    //z AD prevodu do reg PWM
            behADprev = 0;        //nuluje pomocny bit o probihajicim
//prevodu
        }
        if(RA1 == 1)                //pokud je stisknuto tlacitko resetu polohy
            resetPolohy();        //zavola funkci, která resetuje citac polohy
    }
}

//.....
//      Funkce pro vykonani pocatecniho nastaveni uP
//.....
void nastaveni()
{
    //Nastaveni Portu
    CMCON |= 0x111;                //odpojeni komparatoru od pinu na RA
    TRISA = 0b00001111;            //RA4 - RA7 vystup, ostatni piny RA jako dig
                                    //vstupy
    TRISB = 0b00100011;            //RB2, RB3, RB4, RB6 a RB7 jako
                                    //vystupy, ostatni jako vstupy
    RBPU = 1;                      //zakaz rezistoru na RB
    //Nastaveni AD prevodniku
    ANSEL=0b00000001;            //analog vstup je na A0
    ADCON1=0b11000000;            //razeni zprava, casovani od takt frek, ref je
    napajeni
}
```



```

        ADCON0=0b10000001;          //fosc/64, vstup na AN0, nebezi prevod, prevodnik
zapnut
//Nastaveni PWM modulu frekvence PWM bude 19.53kHz
        CCP1CON=0b00001111;          //rezim PWM
        T2CON=0b00000100;            //del pomer preddelicky 1, TMR2 zapnut
        PR2=0xFF;                     //perioda opakovaci frek pwm - FF
//Inicializace pwm modulu
        vystup1 = 0;                  //zapis 0 do reg. PWM
        vystup0 = 0b00001111;         //aby na vystupu PWM byla 0
        CCP1L=vystup1;
        CCP1CON&=vystup0;
//Nastaveni modulu SPI a komunikace
        SSPCON = 0b00100000;          //povoleni ser portu, vys sest hranou, v klidu low,
// MASTER, osc/4
        RB6=1;                        //nastaveni bitu CS
//Nulovani a pocatecni nastaveni promennych
        citac = poc_stav_citace;
        kVyhozeni = 0;
        dataPomoc = 0;
        smer = 0;
        behADprev = 0;
        hodnotaIN = 0;
//Nastaveni prerusovaciho podsystemu na preruseni od nabez hrany na RB0
        INTEDG = 1;                    //preruseni na RB0 od nabezne hrany
        INT0IE = 1;                    //povoleni preruseni od hrany na RB0
        INT0IF = 0;                    //nulovani priznaku preruseni od RB0
        PEIE = 1;                      //povoleni preruseni od periferii
        GIE = 1;                       //povoleni global preruseni
//Nastaveni priznaku potreby resetu
        RA4 = 1;                       //nastaveni bitu signal. potreby kalibrace
}
//.....
//      Funkce pro poslani 12 bitovych dat do DA prevodniku po sbernici SPI
//.....
void posliDoDA(unsigned int data)
{
        RB6 = 0;                       //nulovani bitu CS
        BF = 0;                         //nulovani priznaku plneho zasobniku
        dataPomoc = data >> 8;          //vyber horni casti dat
        SSPBUF =
                (unsigned char)
                (0b00100000 | (0b00001111&(dataPomoc))) ;
                //zapis odesilane hodnoty do zasob
                //a tim zahajeni odesilani dat
        while (BF != 1)                 //cekani na priznak plneho zasob
        {                               //a tim na konec prenosu
        }
        BF = 0;                         //nulovani priznaku plneho zasobniku
        kVyhozeni = SSPBUF;              //nacteni dat ze zasobniku a zahozeni
        SSPBUF =
                (unsigned char) (0x00ff&data) ; //vyber dolni casti dat
        while (BF != 1)                 //cekani na priznak plneho zasob
        {                               //a tim na konec prenosu
        }
        BF = 0;                         //nulovani priznaku plneho zasobniku
        kVyhozeni = SSPBUF;              //nacteni dat ze zasobniku a zahozeni
        RB6 = 1;                        //nastaveni bitu CS do 1
}
//.....
//      Funkce pro obsluhu preruseni, od nabezne hrany na portu RB0
//.....
void interrupt obsluhaPreruseni()
{
        smer = RB5;
        INT0IF = 0;                     //nulovani priznaku preruseni
        if (smer == 1)                   //jeli na RB5 log 1 tzn citat vpred
                citac++;                 //smer kladny - inkrement
        if (smer == 0)                   //jeli na RB5 log 0 tzn citat vzad
                citac--;                 //jinak zaporny - dekrement
}
//.....
//      Funkce pro zapis hodnoty z AD prevodu do modulu PWM
//.....
void zapis_AD_PWM()

```



```
{
    hodnotaIN = ADRESH*0b100000000+ADRESL; //zapise 10bit vysledek do pomoc
                                           //promenne
    vystup0=(unsigned char) (hodnotaIN&0b11); //vybere 2 nejnižsi bity
    vystup0=(vystup0<<4) |0b001111; //posune vlevo
    vystup1=(unsigned char) (hodnotaIN>>2); //vybere 8 vyšších bitů
    CCP1L=vystup1; //zapise obe casti
    CCP1CON=vystup0; //10bit hodnoty do rid
                      //registru
}
//.....
// Funkce pro reset polohy resp. citace, rizená pinem RA3
//.....
void resetPolohy()
{
    __delay_ms(5); //cekani na konec zakmitu
    tlacitka
    if(RA1 == 1) //pokud i potom je log. 1
    pokračuje
    {
        GIE = 0; //zakaz globalního preruseni
        citac = _poc_stav_citace; //nastaveni citace na poc. h.
        while(1)
        {
            if(RA1 == 0) //pokud je tlacitko resetu pustene
            {
                __delay_ms(5); //zpozdeni kvuli zakmitum
                if(RA1 == 0) //pokud i po zakmitech je
                //nulove
                {
                    RA4 =0; //nulovani signalizace
                    INT0IF = 0; //nulovani priznaku
                    //preruseni od RB0
                    GIE = 1; //povoleni global
                    //preruseni
                    break; //konec cyklu while
                }
            }
        }
    }
}
//.....
// Funkce pro prepocet rozsahu polohy
//.....
void prepocetRozsahu(int data)
{
    if(data < 0) //pokud je poloha v dolnim kraji mimo rozsah,
        data = 0; //data jsou minimum rozsahu
    if(data > _maximum_citace) //pokud je poloha v hornim kraji mimo rozsah,
        data = _maximum_citace; //data jsou maximum rozsahu
    float konst = _konst_převodu;
    poloha = (unsigned int) (data*konst); // vrati prepoctenou hodnotu
}

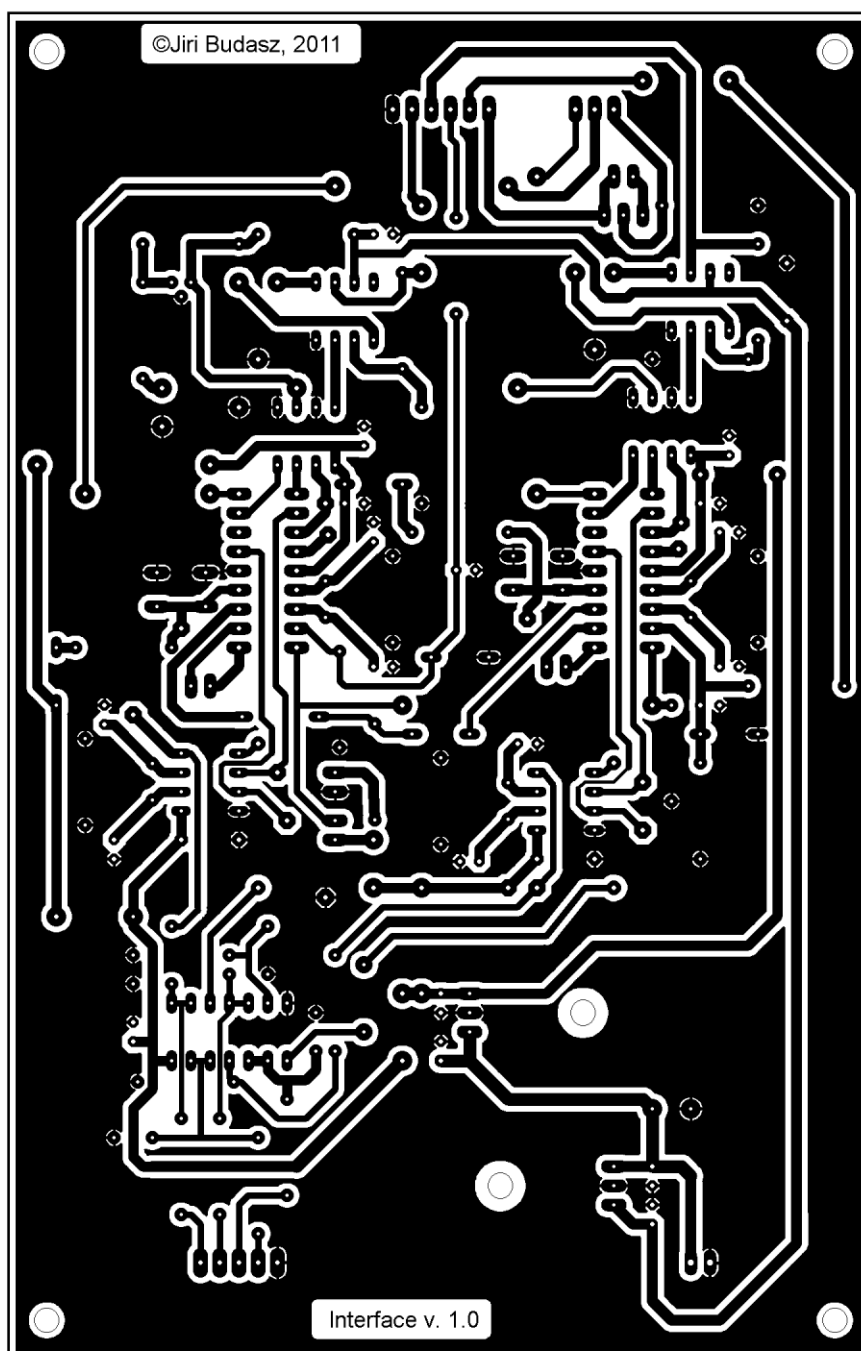
//.....
// Konec kodu
//.....
```



Příloha D - Deska plošných spojů

Motiv spojů – spodní strana

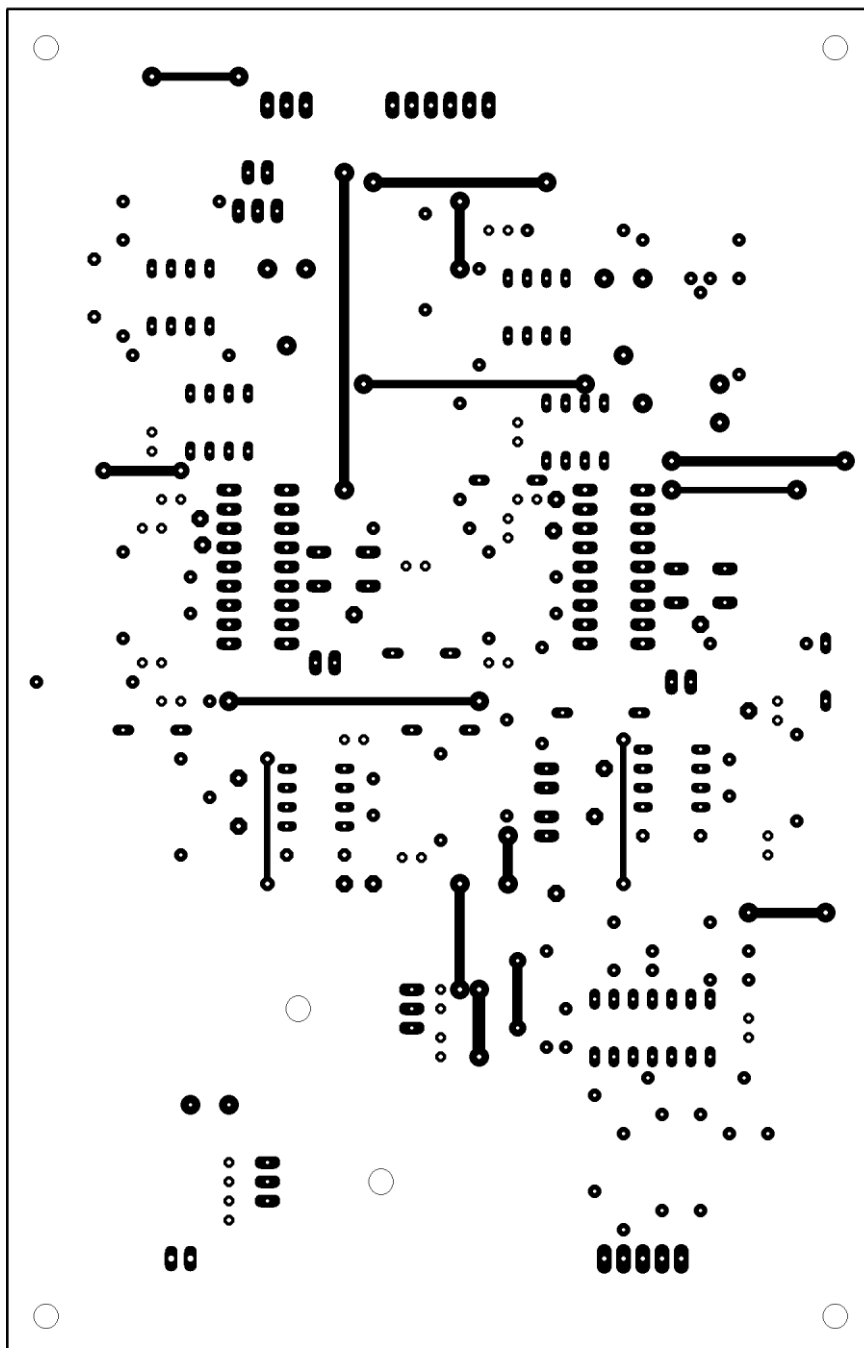
Rozměr desky: 178 mm × 114 mm.



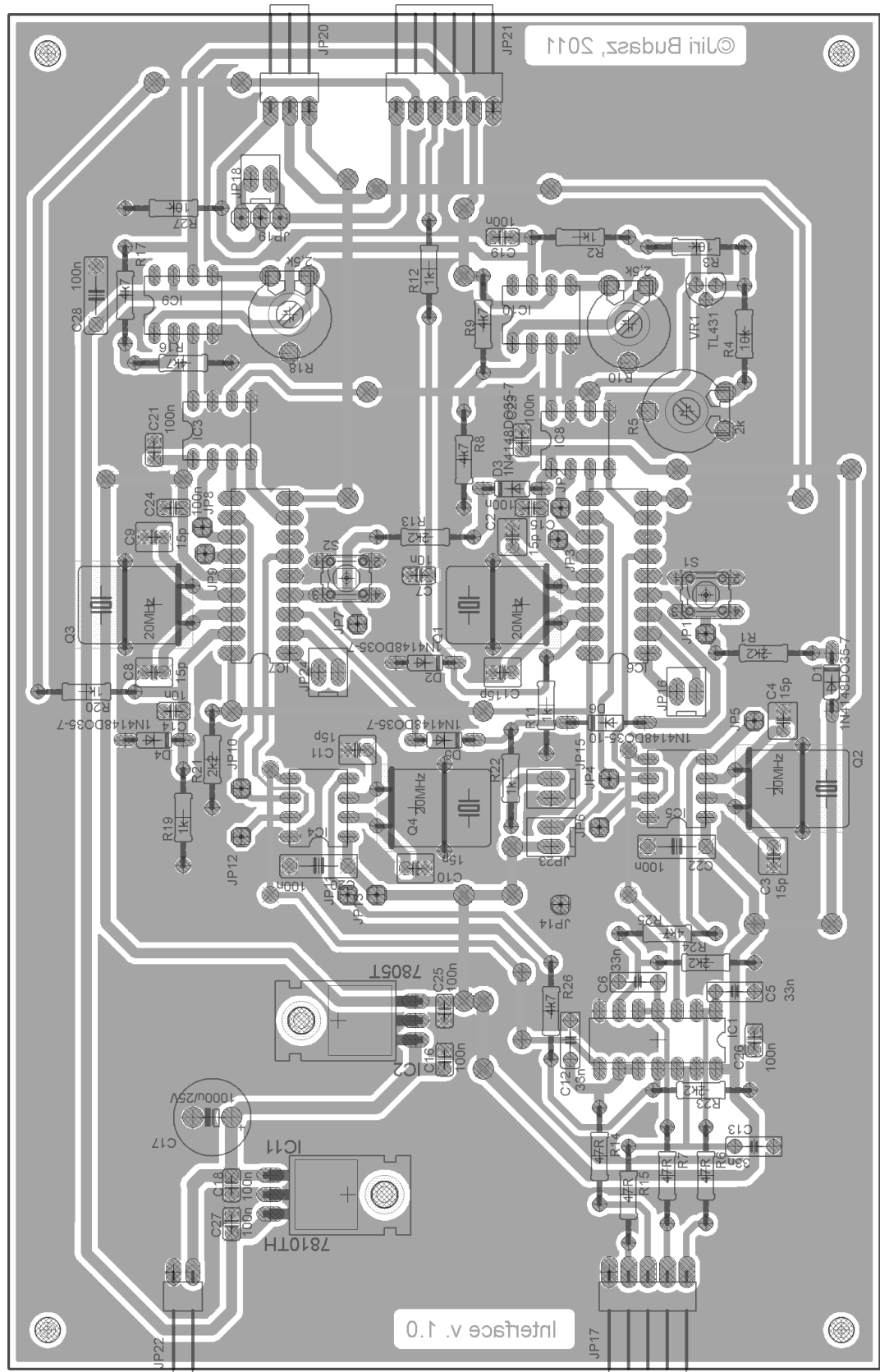


Motiv spojů – vrchní strana

Spoje nahrazeny drátovými propojkami.



Rozmístění součástek – osazovací plán





Seznam součástek

C1	15p	IC2	7805T	Q2	20MHz
C2	15p	IC3	MCP4921-PDIP	Q3	20MHz
C3	15p	IC4	PIC12F629	Q4	20MHz
C4	15p	IC5	PIC12F629	R1	2k2
C5	33n	IC6	PIC16F88	R2	1k
C6	33n	IC7	PIC16F88	R3	10k
C7	10n	IC8	MCP4921-PDIP	R4	10k
C8	15p	IC9	AD820-PDIP	R5	2k
C9	15p	IC10	AD820-PDIP	R6	47R
C10	15p	IC11	7810TH	R7	47R
C11	15p	JP1	1 pin	R8	4k7
C12	33n	JP2	1 pin	R9	4k7
C13	33n	JP3	1 pin	R10	2,5k
C14	10n	JP4	1 pin	R11	1k
C15	100n	JP5	1 pin	R12	1k
C16	100n	JP6	1 pin	R13	2k2
C17	1000u/25V	JP7	1 pin	R14	47R
C18	100n	JP8	1 pin	R15	47R
C19	100n	JP9	1 pin	R16	4k7
C20	100n	JP10	1 pin	R17	4k7
C21	100n	JP11	1 pin	R18	2,5k
C22	100n	JP12	1 pin	R19	1k
C23	100n	JP13	1 pin	R20	1k
C24	100n	JP14	1 pin	R21	2k2
C25	100n	JP15	2 piny	R22	1k
C26	100n	JP16	PC kon2piny	R23	2k2
C27	100n	JP17	PC kon 5pin 90st	R24	2k2
C28	100n	JP18	PC kon 2 piny	R25	4k7
D1	1N4148DO35-7	JP19	3 piny	R26	4k7
D2	1N4148DO35-7	JP20	PC kon 3 piny PC kon 5 pin	R27	10k
D3	1N4148DO35-7	JP21	90st PCKon 2piny	S1	tlacitko do DPS 5 × 5 mm
D4	1N4148DO35-7	JP22	90st	S2	tlacitko do DPS 5 × 5 mm
D5	1N4148DO35-7	JP23	PC kon 2piny		
D6	1N4148DO35-10	JP24	PC kon 2 piny		
IC1	74HC132N	Q1	20MHz	VR1	TL431

Výstupní konektor 37 pinu

Ovladací systém

Signalizací LED

Kalibrační tlačítko

JP15

JP17

JP18

JP20

JP21

JP22

JP23

Propojovací deska

12V

GND

OUT-LED-SIGN

TL-RESET_POLOHY

VCC

IN-PAJENI-12V

GND-PAJENI

OUT-ELEVACE-PWM

OUT-AZIMUT-PWM

OUT-ENABLE

IN-ENABLE

IN-AZIMUT

IN-ELEVACE

IN-AZIMUT-A

IN-ELEVACE-A

IN-AZIMUT-B

IN-ELEVACE-B

IN-AZIMUT-C

IN-ELEVACE-C

IN-AZIMUT-D

IN-ELEVACE-D

IN-AZIMUT-E

IN-ELEVACE-E

IN-AZIMUT-F

IN-ELEVACE-F

IN-AZIMUT-G

IN-ELEVACE-G

IN-AZIMUT-H

IN-ELEVACE-H

IN-AZIMUT-I

IN-ELEVACE-I

IN-AZIMUT-J

IN-ELEVACE-J

IN-AZIMUT-K

IN-ELEVACE-K

IN-AZIMUT-L

IN-ELEVACE-L

IN-AZIMUT-M

IN-ELEVACE-M

IN-AZIMUT-N

IN-ELEVACE-N

IN-AZIMUT-O

IN-ELEVACE-O

IN-AZIMUT-P

IN-ELEVACE-P

IN-AZIMUT-Q

IN-ELEVACE-Q

IN-AZIMUT-R

IN-ELEVACE-R

IN-AZIMUT-S

IN-ELEVACE-S

IN-AZIMUT-T

IN-ELEVACE-T

IN-AZIMUT-U

IN-ELEVACE-U

IN-AZIMUT-V

IN-ELEVACE-V

IN-AZIMUT-W

IN-ELEVACE-W

IN-AZIMUT-X

IN-ELEVACE-X

IN-AZIMUT-Y

IN-ELEVACE-Y

IN-AZIMUT-Z

IN-ELEVACE-Z

IN-AZIMUT-AA

IN-ELEVACE-AA

IN-AZIMUT-AB

IN-ELEVACE-AB

IN-AZIMUT-AC

IN-ELEVACE-AC

IN-AZIMUT-AD

IN-ELEVACE-AD

IN-AZIMUT-AE

IN-ELEVACE-AE

IN-AZIMUT-AF

IN-ELEVACE-AF

IN-AZIMUT-AG

IN-ELEVACE-AG

IN-AZIMUT-AH

IN-ELEVACE-AH

IN-AZIMUT-AI

IN-ELEVACE-AI

IN-AZIMUT-AJ

IN-ELEVACE-AJ

IN-AZIMUT-AL

IN-ELEVACE-AL

IN-AZIMUT-AM

IN-ELEVACE-AM

IN-AZIMUT-AN

IN-ELEVACE-AN

IN-AZIMUT-AO

IN-ELEVACE-AO

IN-AZIMUT-AP

IN-ELEVACE-AP

IN-AZIMUT-AQ

IN-ELEVACE-AQ

IN-AZIMUT-AR

IN-ELEVACE-AR

IN-AZIMUT-AS

IN-ELEVACE-AS

IN-AZIMUT-AT

IN-ELEVACE-AT

IN-AZIMUT-AU

IN-ELEVACE-AU

IN-AZIMUT-AV

IN-ELEVACE-AV

IN-AZIMUT-AW

IN-ELEVACE-AW

IN-AZIMUT-AX

IN-ELEVACE-AX

IN-AZIMUT-AY

IN-ELEVACE-AY

IN-AZIMUT-AZ

IN-ELEVACE-AZ

IN-AZIMUT-BA

IN-ELEVACE-BA

IN-AZIMUT-BB

IN-ELEVACE-BB

IN-AZIMUT-BC

IN-ELEVACE-BC

IN-AZIMUT-BD

IN-ELEVACE-BD

IN-AZIMUT-BE

IN-ELEVACE-BE

IN-AZIMUT-BF

IN-ELEVACE-BF

IN-AZIMUT-BG

IN-ELEVACE-BG

IN-AZIMUT-BH

IN-ELEVACE-BH

IN-AZIMUT-BI

IN-ELEVACE-BI

IN-AZIMUT-BJ

IN-ELEVACE-BJ

IN-AZIMUT-BL

IN-ELEVACE-BL

IN-AZIMUT-BM

IN-ELEVACE-BM

IN-AZIMUT-BN

IN-ELEVACE-BN

IN-AZIMUT-BO

IN-ELEVACE-BO

IN-AZIMUT-BP

IN-ELEVACE-BP

IN-AZIMUT-BQ

IN-ELEVACE-BQ

IN-AZIMUT-BR

IN-ELEVACE-BR

IN-AZIMUT-BS

IN-ELEVACE-BS

IN-AZIMUT-BT

IN-ELEVACE-BT

IN-AZIMUT-BU

IN-ELEVACE-BU

IN-AZIMUT-BV

IN-ELEVACE-BV

IN-AZIMUT-BW

IN-ELEVACE-BW

IN-AZIMUT-BX

IN-ELEVACE-BX

IN-AZIMUT-BY

IN-ELEVACE-BY

IN-AZIMUT-BZ

IN-ELEVACE-BZ

IN-AZIMUT-CA

IN-ELEVACE-CA

IN-AZIMUT-CB

IN-ELEVACE-CB

IN-AZIMUT-CC

IN-ELEVACE-CC

IN-AZIMUT-CD

IN-ELEVACE-CD

IN-AZIMUT-CE

IN-ELEVACE-CE

IN-AZIMUT-CF

IN-ELEVACE-CF

IN-AZIMUT-CG

IN-ELEVACE-CG

IN-AZIMUT-CH

IN-ELEVACE-CH

IN-AZIMUT-CI

IN-ELEVACE-CI

IN-AZIMUT-CJ

IN-ELEVACE-CJ

IN-AZIMUT-CL

IN-ELEVACE-CL

IN-AZIMUT-CM

IN-ELEVACE-CM

IN-AZIMUT-CN

IN-ELEVACE-CN

IN-AZIMUT-CP

IN-ELEVACE-CP

IN-AZIMUT-CQ

IN-ELEVACE-CQ

IN-AZIMUT-CR

IN-ELEVACE-CR

IN-AZIMUT-CS

IN-ELEVACE-CS

IN-AZIMUT-CT

IN-ELEVACE-CT

IN-AZIMUT-CU

IN-ELEVACE-CU

IN-AZIMUT-CV

IN-ELEVACE-CV

IN-AZIMUT-CW

IN-ELEVACE-CW

IN-AZIMUT-CX

IN-ELEVACE-CX

IN-AZIMUT-CY

IN-ELEVACE-CY

IN-AZIMUT-CZ

IN-ELEVACE-CZ

IN-AZIMUT-DA

IN-ELEVACE-DA

IN-AZIMUT-DB

IN-ELEVACE-DB

IN-AZIMUT-DC

IN-ELEVACE-DC

IN-AZIMUT-DD

IN-ELEVACE-DD

IN-AZIMUT-DE

IN-ELEVACE-DE

IN-AZIMUT-DF

IN-ELEVACE-DF

IN-AZIMUT-DG

IN-ELEVACE-DG

IN-AZIMUT-DH

IN-ELEVACE-DH

IN-AZIMUT-DI

IN-ELEVACE-DI

IN-AZIMUT-DJ

IN-ELEVACE-DJ

IN-AZIMUT-DL

IN-ELEVACE-DL

IN-AZIMUT-DM

IN-ELEVACE-DM

IN-AZIMUT-DN

IN-ELEVACE-DN

IN-AZIMUT-DO

IN-ELEVACE-DO

IN-AZIMUT-DP

IN-ELEVACE-DP

IN-AZIMUT-DQ

IN-ELEVACE-DQ

IN-AZIMUT-DR

IN-ELEVACE-DR

IN-AZIMUT-DS

IN-ELEVACE-DS

IN-AZIMUT-DT

IN-ELEVACE-DT

IN-AZIMUT-DU

IN-ELEVACE-DU

IN-AZIMUT-DV

IN-ELEVACE-DV

IN-AZIMUT-DW

IN-ELEVACE-DW

IN-AZIMUT-DX

IN-ELEVACE-DX

IN-AZIMUT-DY

IN-ELEVACE-DY

IN-AZIMUT-DZ

IN-ELEVACE-DZ

IN-AZIMUT-EA

IN-ELEVACE-EA

IN-AZIMUT-EB

IN-ELEVACE-EB

IN-AZIMUT-EC

IN-ELEVACE-EC

IN-AZIMUT-ED

IN-ELEVACE-ED

IN-AZIMUT-EE

IN-ELEVACE-EE

IN-AZIMUT-EF

IN-ELEVACE-EF

IN-AZIMUT-EG

IN-ELEVACE-EG

IN-AZIMUT-EH

IN-ELEVACE-EH

IN-AZIMUT-EI

IN-ELEVACE-EI

IN-AZIMUT-EJ

IN-ELEVACE-EJ

IN-AZIMUT-EL

IN-ELEVACE-EL

IN-AZIMUT-EM

IN-ELEVACE-EM

IN-AZIMUT-EN

IN-ELEVACE-EN

IN-AZIMUT-EO

IN-ELEVACE-EO

IN-AZIMUT-EP

IN-ELEVACE-EP

IN-AZIMUT-EQ

IN-ELEVACE-EQ

IN-AZIMUT-ER

IN-ELEVACE-ER

IN-AZIMUT-ES

IN-ELEVACE-ES

IN-AZIMUT-ET

IN-ELEVACE-ET

IN-AZIMUT-EU

IN-ELEVACE-EU

IN-AZIMUT-EV

IN-ELEVACE-EV

IN-AZIMUT-EW

IN-ELEVACE-EW

IN-AZIMUT-EX

IN-ELEVACE-EX

IN-AZIMUT-EY

IN-ELEVACE-EY

IN-AZIMUT-EZ

IN-ELEVACE-EZ



Příloha F - Tabulky z měření

Tabulky pro měření převodní charakteristiky odměřování

Azimut			
úhel [°]	U [V]	úhel [°]	U [V]
-130	0,0043	10	5,314
-120	0,3976	20	5,694
-110	0,783	30	6,06
-100	1,174	40	6,42
-90	1,572	50	6,79
-80	1,940	60	7,15
-70	2,313	70	7,53
-60	2,710	80	7,91
-50	3,076	90	8,28
-40	3,462	100	8,66
-30	3,833	110	9,04
-20	4,205	120	9,41
-10	4,597	128	9,71
0	4,972		
Voltmetr: Pro'sKit MT-1860 v.č. 992958921 Úhloměr: rýsovací, průměr 100 mm, rozlišení 1°			

Elevace			
úhel [°]	U [V]	úhel [°]	U [V]
-49	0,0047	0	4,995
-45	0,2096	5	5,574
-40	0,853	10	6,11
-35	1,331	15	6,62
-30	1,855	20	7,13
-25	2,351	25	7,7
-20	2,881	30	8,2
-15	3,411	35	8
-10	3,921	40	74
-5	4,434	45	9,71
Voltmetr: Pro'sKit MT-1860 v.č. 992958921 Úhloměr: rýsovací, průměr 100 mm, rozlišení 1°			



Tabulka pro měření převodu napětí na PWM signál

Azimut		Elevace	
U [V]	střída[%]	U [V]	střída[%]
0	0	0	0
0,5579	4	0,5293	4
1,08	10	1,042	8
1,592	16	1,529	16
2,013	19,5	2,075	20
2,554	25	2,570	24
3,042	29,7	3,021	28
3,513	34,4	3,516	36
4,030	39,1	4,067	40
4,526	43,8	4,567	44
5,016	49,2	5,073	52
5,529	54,7	5,518	56
6,01	59,4	6,021	60
6,52	64,1	6,55	64
7,16	70,3	7,02	68,8
7,55	75	7,54	74,2
8,04	79,7	8,11	79,7
8,59	85,2	8,58	84,4
9,11	90,6	9,26	91,4
9,58	95,3	9,69	95,3
9,83	96,9	9,75	96,9
Voltmetr: Pro'sKit MT-1860 v.č. 992958921 Osciloskop: RIGOL DS1102D		Voltmetr: Pro'sKit MT-1860 v.č. 992958921 Osciloskop: RIGOL DS1102D	